2008

Non-Unique oligonucleotide probe selection heuristics

Lili Wang
University of Windsor

NON-UNIQUE OLIGONUCLEOTIDE PROBE SELECTION HEURISTICS

BY

LILI WANG

FACULTY OF GRADUATE STUDIES
UNIVERSITY OF WINDSOR
2008

Non-Unique Oligonucleotide Probe Selection Heuristics

by

Lili Wang

A Thesis
Submitted to the Faculty of Graduate Studies
through Computer Science
in partial fulfillment of the requirements for
the Degree of Master of Science at the
University of Windsor

Windsor, Ontario, Canada
2008

Non-Unique Oligonucleotide Probe Selection Heuristics

by

Lili Wang

APPROVED BY:

J. Hudson, External Reader
Department of Biology

R. Gras, Internal Reader
School of Computer Science

For Dr. Ngom.

A. Ngom, Advisor
School of Computer Science

D. Wu, Chair of Defense
School of Computer Science

10 September 2008

Declaration of Co-Authorship / Previous Publication

I. Co-Authorship Declaration

I hereby declare that this thesis incorporates material that is result of joint research, as follows:
This thesis also incorporates the outcome of a joint research undertaken in collaboration with professor Dr. Robin Gras and Dr. Luis Rueda. The collaboration is covered in Chapter 3 and Chapter 4 of the thesis. In all cases, the key ideas, primary contributions, experimental designs, data analysis and interpretation, were performed by the author, and the contribution of co-authors was primarily through the provision of corrections and constructive criticism.

I am aware of the University of Windsor Senate Policy on Authorship and I certify that I have properly acknowledged the contribution of other researchers to my thesis, and have obtained written permission from each of the co-author(s) to include the above material(s) in my thesis.

I certify that, with the above qualification, this thesis, and the research to which it refers, is the product of my own work.

II. Declaration of Previous Publication

This thesis includes *5* original papers that have been previously published/submitted for publication in peer reviewed journals, as follows:

Thesis Chapter	Publication title/full citation	Publication status*
Chapter 3	***Wang, L., Ngom, A., and Rueda, L.*** *2008. Sequential forward selection approach to the non-unique oligonucleotide probe selection problem. In Proceedings of the third IAPR International Conference on Pattern Recognition in Bioinformatics, Melbourne, Australia.* ***Wang, L. and Ngom, A.*** *2007. A model-based approach to the non-unique oligonucleotide probe selection problem, In Proceedings of the Second International Conference on Bio-Inspired Models of Network, Informatiaon, and Computing Systems(Bionetics 2007), Budapest, Hungary, ISBN:978-963-9799-05-9.*	*accepted for publication* *published*
Chapter 4	***Wang, L., Ngom, A., Gras, R., and Rueda, L.*** *2008. An evolutionary approach to the non-unique oligonuleotide probe selection problem, Springer Transactions on Computational System Biology.*	*in press*

	Wang, L., Ngom, A., Gras, R., and Rueda, L. *2008. Evolution strategy with greedy probe selection heuristics for the non-unique oligonucleotide probe selection problem. In Proceedings of the 2008 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, Sep.15-17, Sun Valley, Idaho, USA.*	*accepted for publication*
	Wang, L., Ngom, A., and Gras, R. *2008. Non-unique oligonucleotide microarray probe selection method based on genetic algorithm. In Proceedings of the 2008 IEEE Congress on Evolutionary Computation, Hong Kong, China, 1004-1011.*	*published*

I certify that I have obtained a written permission from the copyright owner(s) to include the above published material(s) in my thesis. I certify that the above material describes work completed during my registration as graduate student at the University of Windsor.

I declare that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University of Institution.

ABSTRACT

The *non-unique probe selection problem* consists of selecting both *unique* and *non-unique* oligonucleotide probes for oligonucleotide microarrays, which are widely used tools to identify viruses or bacteria in biological samples. The *non-unique* probes, designed to hybridize to at least one target, are used as alternatives when the design of *unique* probes is particularly difficult for the closely related target genes. The goal of the *non-unique probe selection problem* is to determine a smallest set of probes able to identify all targets present in a biological sample. This problem is known to be NP-hard. In this thesis, several novel heuristics are presented based on greedy strategy, genetic algorithms and evolutionary strategy respectively for the minimization problem arisen from the *non-unique probe selection* using the best-known ILP formulation. Experiment results show that our methods are capable of reducing the number of probes required over the state-of-the-art methods.

DEDICATION

To my family

for their endless understanding, encouragement and love

...

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my family. Without their support and love, it would not have been possible for me to pursue great things in my life.

I would like to express my deepest appreciation to my advisor, Dr. Alioune Ngom, for his encouragement, support and invaluable suggestions in guiding me towards the successful completion of this research work. Without his generous funding and advice, it would be hard for me to achieve so many publications in this work.

I would like to express my great gratitude to Dr. John W. Hudson, Department of Biology, and Dr. Robin Gras, School of Computer Science for giving me corrections and constructive criticism to improve the quality of this research, for their patience in arranging the time of my proposal and defence, and for being in the committee, and to Dr. Dan Wu for serving as the chair of the committee.

I gratefully acknowledge the assistance of Dr. Panos M. Pardalos and Dr. Michelle A. Ragle, Department of Industrial and Systems Engineering, University of Florida, for providing all the data sets used in this thesis.

Finally I want to extend my gratitude to my friends, the faculty members and staff of the School of Computer Science for their friendly suggestions and support during my study at University of Windsor.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

CHAPTER I

INTRODUCTION

Oligonucleotide microarrays are widely used tools, in molecular biology providing a fast and cost-effective method for monitoring the expression of thousands of genes simultaneously [32]. In order to measure the expression level of a specific gene in a sample, one must design a microarray containing short strands of known DNA sequences of 8 to 30 bp, called *oligonucleotide probes*, which are complementary to the gene's segments, called *targets*. These targets, if present in the sample, should bind to their complementary probes by means of *hybridization*. Typically, the total length of a probe used to hybridize a gene is only a small fraction of the length of the gene [32]. The success of a microarray experiment depends on how well each probe hybridizes to its target. Expression levels can only be accurately measured if each probe hybridizes to its target only, given the target is present in the biological sample at any concentration. However, choosing good probes is a difficult task since different sequences have different hybridization characteristics.

A probe is *unique*, if it is designed to hybridize to a single target. However, due to hybridization errors, there is no guarantee that unique probes will hybridize to their intended targets only. Many parameters such as secondary structure, salt concentration, GC content, free energy and melting temperature also affect the hybridization quality of probes [32], and their values must be carefully determined to design high quality probes. It is particularly difficult to design unique probes for closely related genes that are to be identified. Too many targets will be similar and hence hybridiza-

tion errors increase substantially. An alternative approach is to devise a method that can make use of *non-unique* probes, i.e. probes that are designed to hybridize to at least one target [32]. The *non-unique probe selection problem* is to determine a smallest set of probes able to identify all targets present in a biological sample. This is proven an NP-hard problem [19]. Some fundamental questions will be addressed firstly, before stating the non-unique probe selection problem in this section.

I-1 Functional Genomics

Functional genomics attempt to describe gene or protein functions and interactions by the usage of vast data produced by genomic projects, such as genome sequencing projects. Functional genomics includes function-related aspects of the genome such as mutation and polymorphism analysis, as well as measurement of molecular activities [47].

Functional genomics uses mostly high-throughput techniques to characterize the abundance gene products such as DNA microarrays and serial analysis of gene expression (SAGE) for mRNA; two-dimensional gel electrophoresis and mass spectrometry for protein. More detailed descriptions can be found in [30].

I-2 Microarray Analysis

The foundation of microarray technology lies in the Watson-Crick complementarity of double-stranded DNA or RNA-DNA-hyrids [30]. DNA forms a double-helix and

consists of two antiparallel complementary strands. Each strand is a directional linear polymer of four types of nucleotides or bases (adenine A, cytosine C, guanine G, and thymine T), held by a sugar-phosphate backbone. RNA occurs as a single-stranded molecule with four types of bases (A, C, G, and uracil U). Figure 1 shows the DNA and RNA structure.

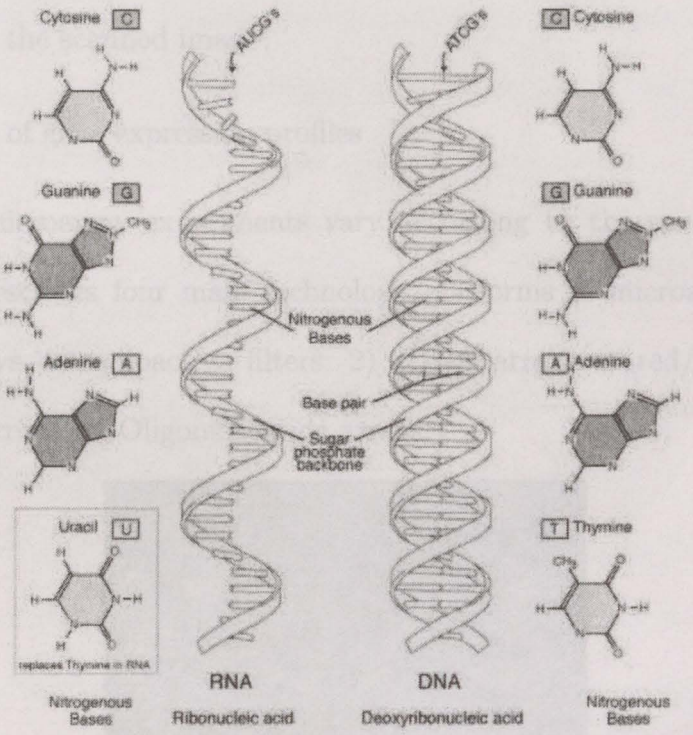


Figure 1: DNA and RNA structure. Image cited from [30], p.2

Microarray technology utilizes nucleic acid hybridization techniques and computing technology to evaluate the expression profile of thousands of genes within a single experiment. It has been proven to be an extremely powerful tool to efficiently utilize the enormous amount of information provided by the completion of numerous genome projects. A typical gene expression microarray experiment involves the fol-

lowing steps:

1. Target preparation
2. Hybridization
3. Washing, staining, and scanning of the array
4. Analysis of the scanned image
5. Generation of gene expression profiles

The details of microarray experiments vary according to the specific type of microarray. [30] describes four main technology platforms of microarrays: 1) Nylon membrance arrays or radioactive filters; 2) cDNA arrays or red/green arrays; 3) Polynucleotide arrays; 4) Oligonucleotide arrays.

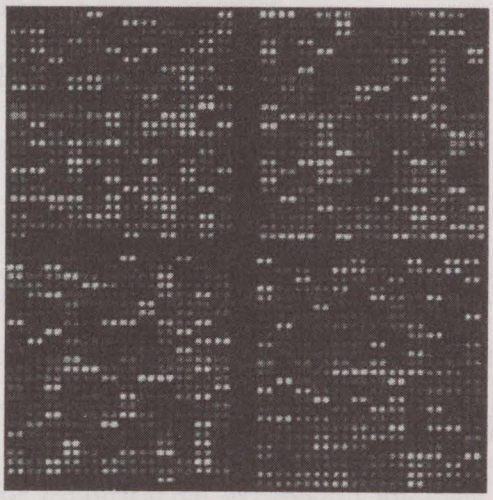


Figure 2: DNA microarray

In Oligonucleotide arrays (also called DNA chips), shown in Figure 2, oligonucleotides, usually 25-mers, are directly synthesized onto a glass wafer by a combination

of semiconductor-based photolithography and solid phase chemical synthesis technologies. Each array contains up to 900,000 different oligos and each oligo is present in millions of copies. Since oligonucleotide probes are synthesized in known locations on the array, the hybridization patterns and signal intensities can be interpreted in terms of gene identity and relative expression levels. Diamandis [10] discussed the microarray technology as a powerful tool for molecular diagnostics. Couzinet *et al.* [5] evaluated the ability of a high-density DNA probe array based on 16S rDNA sequences to identify Staphylococcus species.

I-3 Unique Probe Selection

Oligonucleotide probe is a fragment of DNA used to detect the presence of nucleotide sequences (targets) in DNA or RNA samples.

The unique probe selection problem, also called probe design problem, is defined as: Given a set of targets $T = (t_1, \ldots, t_n)$ and a parameter m which specifies the length of the probes, the probe design problem finds, for every target t_i, a length-m probe, which satisfies (1) Homogeneity, (2) Sensitivity and (3) Specificity [36].

In [30], the unique probe selection problem is formulated as: Given hybridization parameters θ and a set of target sequences $T = (t_1, \ldots, t_n)$, design a set of unique probes for each target for quantitative expression analysis. The hybridization parameters θ take account of temperature, salt concentration, number and density of probe molecules on the probe's spot, cRNA fragment length distribution, and other conditions specified in experimental protocols [30].

A probe is called *unique* if it hybridizes to its intended target only, under specified experimental conditions [32]. The high degree of similarity in large families of closely related target sequences makes it impossible to find one unique probe for every target, given the probe length and melting temperature constraints. In some cases on robust presence or absence calls, such as in virus subtyping, unique probes are not a necessity[32]. An alternative approach is to devise a method that can make use of *non-unique* probes. In [30], the criteria for probe set selection is also described. In this thesis, we focus on the *non-unique* probe selection problem, which is a totally different optimization problem from the *unique* probe selection.

I-4 Non-Unique Probe Selection

The *non-unique probe selection problem* is to determine a smallest set of probes able to identify all targets present in a biological sample. This is proved to be an NP-hard problem [19].

Given a target set $T = \{t_1, \ldots, t_m\}$, and probe set $P = \{p_1, \ldots, p_n\}$, an $m \times n$ *target-probe incidence matrix* $H = [h_{ij}]$ is such that $h_{ij} = 1$, if probe p_j hybridizes to target t_i, and $h_{ij} = 0$ otherwise. Table 1 shows an example of a matrix with $m = 4$ targets and $n = 6$ probes. A probe p_j *separates* two targets, t_i and t_k, if it is a substring of either t_i or t_k, that is, if $|h_{ij} - h_{kj}| = 1$. For example, if t_i = AGGCAATT and t_k = CCATATTGG, then probe p_j = GCAA separates t_i and t_k, since it is a substring of t_i only, whereas probe p_l = ATT does not separate t_i and t_k, since it is a substring of both targets [23]. Two targets, t_i and t_k, are *s-separated*, $s \geq 1$, if

there exist at least s probes such that each separates t_i and t_k; in other words, the Hamming distance between rows i and k in H is at least s. For example, in Table 1 targets t_2 and t_4 are 4-separated. A target t is *c-covered*, $c \geq 1$, if there exist at least c probes such that each hybridizes to t. In Table 1, target t_2 is 3-covered. Due to hybridization errors in microarray experiments, it is required that any two targets be $s_{\min}$-separated and any target be $c_{\min}$-covered; usually, we have $s_{\min} \geq 2$ and $c_{\min} \geq 2$. These two requirements are called *separation constraints* and *coverage constraints.*

Table 1: A 4×6 target-probe incidence matrix.

	p_1	p_2	p_3	p_4	p_5	p_6
t_1	1	1	0	1	0	1
t_2	1	0	1	0	0	1
t_3	0	1	1	1	1	1
t_4	0	0	1	1	1	0

Given a matrix H, the aim of the non-unique probe selection problem is to find a minimal probe set that determines the presence or absence of specified targets, and such that all constraints are satisfied. In Table 1, if $s_{\min} = c_{\min} = 1$ and assuming that exactly one of $t_1, \ldots, t_4$ is in the sample, then the goal is to select a minimal set of probes that allows us to infer the presence or absence of a single target. In this case, a minimal solution is $\{p_1, p_2, p_3\}$ since for target t_1, probes p_1 and p_2 hybridize while p_3 does not; for target t_2, probes p_1 and p_3 hybridize while p_2 does not; for target t_3, probes p_2 and p_3 hybridize while p_1 does not; and finally for target t_4, only

probe p_3 hybridize. Thus, each single target will be identified by the set $\{p_1, p_2, p_3\}$, if it is the only target present in the sample; moreover, all constraints are satisfied. For $s_{\min} = c_{\min} = 2$, a minimal solution that satisfies all constraints is $\{p_2, p_3, p_5, p_6\}$. Of course, $\{p_1, \ldots, p_6\}$ is a solution but it is not minimal, and hence is not cost-effective.

Stated formally, given an $m \times n$ matrix H with a target set $T = \{t_1, \ldots, t_m\}$ and a probe set $P = \{p_1, \ldots, p_n\}$, and a minimum coverage parameter $c_{\min}$, a minimum separation parameter $s_{\min}$ and a parameter $d_{\max} \geq 1$, the aim of the non-unique probe selection problem is to determine a subset $P_{\min} = \{q_1, q_2, \cdots, q_s\} \subseteq P$ such that:

1. $s = |P_{\min}| \leq n$ is minimal.
2. Each target $t_i \in T$ is $c_{\min}$-covered by some probes in $P_{\min}$.
3. Each target-pair $(t_i, t_k) \in T \times T$ is $s_{\min}$-separated by some probes in $P_{\min}$.
4. Each pair of small groups of targets ($\leq d_{max}$) is $s_{\min}$-separated by some probes in $P_{\min}$.

This problem was proved to be NP-hard in [19], by performing a reduction from the *set covering problem.* It is NP-hard even for $c_{\min} = 1$ or $s_{\min} = 1$. The work of [18] and [19] formulated the non-unique probe selection problem as an *integer linear programming* (ILP) problem. Let $x_j (1 \leq j \leq n)$ be the set of binary variables with $x_j = 1$ if probe p_j is chosen and 0 otherwise. We have:

$$\text{Minimize: } \sum_{j=1}^{n} x_j \ . \tag{1}$$

Subject to:

$$x_j \in \{0,1\} \qquad 1 \leq j \leq n \ , \tag{2}$$

$$\sum_{j=1}^{n} h_{ij} x_j \geq c_{\min} \qquad 1 \leq i \leq m \ , \tag{3}$$

$$\sum_{j=1}^{n} |h_{ij} - h_{kj}| \, x_j \geq s_{\min} \qquad 1 \leq i < k \leq m \ . \tag{4}$$

Function (1) minimizes the number of probes. The probe selection variables are binary-valued in Restriction (2). Constraints (3) and (4) are the coverage and separation constraints, respectively. Note that Constraints (4) are for single targets only. [19] proposed the following ILP formulation that also includes the group separation constraints for aggregated targets:

$$\text{Minimize: } \sum_{j=1}^{n} x_j \ . \tag{5}$$

Subject to:

$$x_j \in \{0,1\} \qquad 1 \leq j \leq n \ , \tag{6}$$

$$\sum_{j=1}^{n} \left| \omega_j^{t_x^{\mathrm{a}}} - \omega_j^{t_y^{\mathrm{a}}} \right| x_j \geq \min \left\{ d, \sum_{j=1}^{n} \left| \omega_j^{t_x^{\mathrm{a}}} - \omega_j^{t_y^{\mathrm{a}}} \right| \right\} \qquad \forall (t_x^{\mathrm{a}}, t_y^{\mathrm{a}}) \in 2^T \times 2^T \ , \tag{7}$$

$$|t_x^{\mathrm{a}}|, |t_y^{\mathrm{a}}| \leq d_{\max} \ ,$$

$$t_x^{\mathrm{a}} \neq t_y^{\mathrm{a}} \ .$$

where $c_{\min} = s_{\min} = d$. Here, Constraints (7) are the group separation constraints which also contain the single target separation constraints. The coverage constraints are also satisfied by Equation 7 with $t_x^{\mathrm{a}} = \emptyset$ and $t_y^{\mathrm{a}} = \{t_i\}$ for $1 \leq i \leq m$.

In this thesis, we proposed several heuristics to solve the ILP formulation (Equation 1). Note that one can easily check if the probes in the original set of candidate

satisfy all the constraints. If not, then there are no feasible solutions. In this case, we can insert *unique virtual probes* in the original probe set only for those targets or target-pairs that are not $c_{\min}$-covered or $s_{\min}$-separated. This will ensure the existence of feasible solutions.

I-5 Contribution

In this thesis, several heuristics will be proposed based on greedy strategy and evolutionary approaches respectively, for the minimization problem arisen from non-unique probe selection using the ILP formulation for single target only (Equation 1). The Greedy Heuristics presented include:

1. Dominated Row Covering Heuristic (DRC)
2. Dominated Probe Selection Heuristic (DPS)
3. Normalized Dominant Probe Selection Heuristic (DPSn)
4. Dynamic DRC, DPS and DPSn Heuristics
5. Sequential Forward Probe Selection Algorithm (SFPS)

This thesis contributes the first evolutionary approaches for solving this minimization problem. Evolutionary Heuristics:

1. Genetic Algorithm with DRC Heuristic
2. Evolution Strategy with DDRC and DDPS

I-6 Thesis Organization

The thesis is organized in six chapters. Chapter II provides a survey of unique probe selection and non-unique probe selection. Chapter III presents the proposed deterministic greedy heuristics for non-unique probe selection problem. Chapter IV presents the proposed Genetic Algorithm and evolutionary strategy for non-unique probe selection problem. Chapter V deals with experiment results and performance analysis, where all proposed approaches are analyzed and compared to current published methods. Finally, Chapter VI concludes the thesis and identified open research problems arising from this work.

CHAPTER II

REVIEW OF LITERATURE

II-1 Unique Probe Selection Problem

The simple approach for probe selection problem would be to use random oligonucleotides. However, DNA sequences are not really random in nature, so a random probe is not likely to occur in a sufficient number of clones to provide adequate discrimination [3]. Due to its significance, probe selection attracts a lot of attention. Various probe selection approaches have been developed. In [6], Cutichia *et al.* provided a methodology for choosing synthetic oligonucleotide probes to be used in contig mapping experiments, based on constraints with respect to frequency of occurrence within a particular genome and the G + C content.

Li and Stormo [20] developed a heuristic approach to optimize the selection of specific probes for each gene in an entire genome based on the free energy and melting temperature criteria. They stated that the optimized probes for each gene provided more accurate determinations of true expression levels by minimizing background hybridization, and eliminating the need for multiple probes per gene.

The probe selection had been formulated as an explicit optimization problem in [15]. Herwig *et al.*[15] presented an information theoretical probe selection approach, which is a greedy heuristic based on clustering and entropy. They stated that their approach was superior to the selection of probes according to their frequencies, and to randomly chosen probe sets [15].

Tobler *et al.*[38] empirically evaluated three standard machine learning algorithms: naive Bayes, decision trees and artificial neural networks in the task of predicting good probes. As a result, two of the learning algorithms, naive Bayes and neural networks, learnt to predict probe quality surprisingly well, but decision tree induction and the simple approach of using predicted melting temperature to rank probes performed significantly worse than those two learning algorithms [38]. By the way, they also stated that the nucleotides in the middle of the probes sequence were more informative than those at the ends of the sequence [38].

Rahmann [26] presented the first algorithm selecting oligonucleotide probes for microarray experiments on a large scale. This algorithm based on a suffix array with additional information that is efficient both in terms of memory usage and running time to rank all candidate oligos according to their specificity [26]. Later, in [27] Rahmann proposed the longest common factor approach for large scale oligonucleotide selection. In [28], Rahmann contributed an approach using the concept of jumps to improve the accuracy of the longest common factor approach for probe selection by moving from a string-based to an energy-based specificity measure.

Wang *et al.* [46] presented a strategy for picking oligos for microarrays that focus on a design universe consisting exclusively of protein coding regions. In [46], they discussed the oligo picking criteria, such as location in the sequence, T_m uniformity, probe accessibility, reduced cross-hybridization, and evasion of non-coding RNA and low complexity regions. In their experiments, sequences that had no unique probes were represented by non-unique probes.

Sung *et al.* [36] presented a fast and accurate probe selection algorithm for large

genomes. In [34], Shin *et al.* proposed a probe design approach using ε-multi-objective evolutionary algorithms with thermodynamic criteria. Tulpan [39] introduced new algorithms for design of DNA strand sets that satisfy any of several combinatorial and thermodynamic constraints.

II-2 Non-Unique Probe Selection Problem

The first work about non-unique probe selection problem was due to Boreman *et al.*[3]. In [3], Boreman *et al.* introduce two alternative formulations of probe selection, Minimum Cost Probe Set (MCPS) and Maximum Distinguishing Probe Set (MDPS). The Minimum Cost Probe Set problem is a special case of the non-unique probe selection problem with both c_{min} and s_{min} set to 1. The Maximum Distinguishing Probe Set problem consists of finding a set of k probes that maximizes the number of distinguished pairs of clones. Both MCPS and MDPS problems are variants of *Set Cover Problem* and are NP-hard [3]. Borneman *et al.* [3] proposed two efficient heuristics for minimizing the number of oligonucleotide probes for analyzing populations of ribosomal RNA gene (rDNA) clones by hybridization experiments on DNA microarrays, based on simulated annealing for MDPS and Lagrangian relaxation for MCPS.

Rash and Gusfield [31] considered the minimum cost probe set problem using suffix trees. The approach starts with a set of known strings (viruses) and builds a minimum cardinality set of substrings, which is adequate to identify an unknown string using substring tests. In this approach, suffix trees are used to reduce the number of variables in an ILP formulation. They state three key technical ideas in

their basic implementation: the use of suffix trees to identify the critical substring, ILP to express the minimization problem, and reduction in the size of the ILP [31]. Rash and Gusfield also extended their basic implementation to deal with mutations and sequencing errors by adding minimum separation constraints to the ILP.

In [32], Schliep *et al.* proposed a statistical, non-adaptive group testing scheme for the microarray setting. In this approach, the target sequences correspond to individuals, potential groups are specified by a probe, which hybridizes to a set of target sequences, and the goal is to devise a group testing design which covers each target with a certain number of probes and allows identification of several targets simultaneously [32]. The cross-hybridization and error tolerance were token into account explicitly, compared with previous work in [3] and [31].

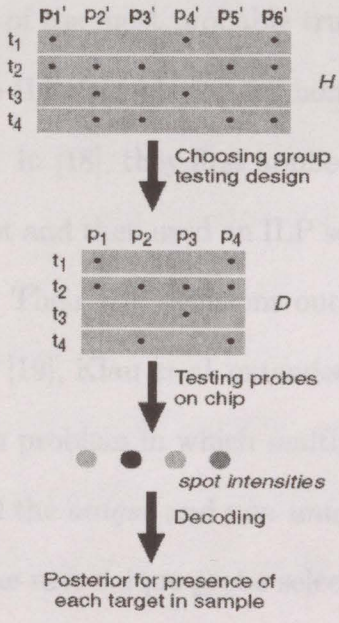


Figure 3: An overview of the group testing approach in [32]

Like in Figure 3, the whole procedure in this approach can be summarized as follows:

1. Collect suitable probe candidates.
2. From those candidates, find a minimum subset of probes that allows discrimination between as many target sets as possible.
3. Decode the presence or absence of target sequences.

For the step 2 above, Schliep *et al.* [32] described a simple but fast greedy heuristic which computes an approximate solution that guarantees $s_{\min}$-separation for pairs of small aggregated targets. Once the hybridization experiment was performed, a Markov Chain Monte Carlo approach for the decoding was applied and the result of the decoding was a sorted list of the most probable true-positive targets.

Klau *et al.* [18] stated the ILP formulation for non-unique probe selection problem, but for single target only. In [18], they first applied a greedy heuristic to reduce the original candidate probe set and then used an ILP solver such as CPLEX software to further reduced the result. Their ILP solutions outperformed those of [32] in all instances. In subsequent work [19], Klau *et al.* extended their ILP formula of [18] for the non-unique probe selection problem in which multiple targets may be present.

In [30], Rahmann explained the *unique* and *non-unique* probe selection problem in detail. Rahmann [29] stated the non-unique probe selection problem as the condition optimization problem. In [29], Rahmann proposed a greedy heuristic to select an appropriate subset of probes, given many potential probe candidates and the target-probe incidence matrix. This heuristic started with a full design and iteratively

removed a single row to locally minimize the condition. Rahmann claimed that although the greedy heuristic did not always find the optimal solution, its performance was reasonably close to the optimal design and much better than choosing random subsets[29]. In [30], Rahmann described a statistical group testing approach for non-unique probe selection problem. Within this approach, a fast heuristic to find a good group testing design D to select rows of the the full $m \times n$ probe-target hybridization matrix H, and an optimal design method based on integer linear programming (ILP) are presented.

Gąsieniec *et al.* [12] proposed a new direction to tackle the probe selection for DNA microarrays. They focused on the efficient selection of a minimal set of probes, and used a limited number of non-unique probes in the context of a large family of closely homologous genes. Their approach took a set of known gene sequences as input and built a small cardinality set of probes allowing to identify the unknown target in the sample. Instead of checking all possible probes, they exploited randomization. They randomly pick probes with some minimal criteria checking. Their experimental results showed that almost all genes could be uniquely identified by a single probe; the others need at most a combination of two probes [12].

In previous work [31][32][18][29][30][12], only the ability to detect *known* targets has been evaluated, so Schliep *et al.* [33] extended the group testing approach using non-unique probes to targets related by a phylogenetic tree, the first work to address detecting the presence of yet *unknown* targets.

Moreover, group testing approaches have been discussed by [40][8][37][9].

Wang *et al.* [40] gave an theoretical overview on the group testing methods for the

non-unique probe selection problem, and showed that when every probe hybridizes to at most two targets, the minimization is still MAX SNP-complete, but has a polynomial-time approximation with performance ratio $1 + \frac{2}{d+1}$ [40].

Deng *et al.* [8] described the non-adaptive group testing approach for the non-unique probe selection problem, and gave a mini survey on the computational complexity and approximation algorithms for the minimization problem. They claimed that the best known design of non-adaptive group testing was within a factor of $O(\log d)$ from the lower bound and the best known approximation for the non-unique probe selection is within a factor of $O(\log n)$ from optimal solution [8].

Thai *et al.* [37] present a novel decoding algorithm identifying all positive clones in the presence of inhibitors and experimental errors for the pooling design. The pooling design is also called non-adaptive group testing, which is a mathematical tool to significantly reduce the number of tests in DNA library screening. In DNA library screening, the basic problem of group testing is to identify the set of all positive clones in a large population of clones with the minimum number of tests [37].

In [8] and [37], the authors did not provide any practical approach, and only theoretical results had been discussed. In 2008, Deng *et al.* [9] extended their research and proposed efficient algorithms based on Integer Linear Programming to select a minimum number of non-unique probes using d-disjunct matrices. In [9], they constructed a d-disjunct matrix instead of a d-separable matrix considering the computational complexity of decoding. Deng *et al.* improved the decoding complexity compared with the approach in [19]. The decoding complexity of their algorithms was claimed to be $O(n)$ to identify up to d targets with error tolerance [19].

Based on the same ILP formulation (Equation 1), the efficient computation of the minimum set of candidate probes with the minimum coverage and separation constraints, given a target set T, probe set P, and the target-probe incidence matrix H, has been paid more attention by [23] and [25] recently.

Meneses *et al.* [23] proposed a greedy non-random heuristic for the non-unique probe selection problem, based on ILP formula [18], for single target only. They first used local search and sorting to construct a feasible solution to the ILP, and then further reduced this set by iteratively removing probes in such a way that the coverage and separation constraints were still satisfied. Meneses [23] tested their algorithm on the data used in [18]. The algorithm greatly outperformed the ILP method of [18] for the largest and only real-world dataset, although the solutions for the smaller, artificial datasets contained more probes than those found in [18].

Ragle *et al.* [25] developed an *optimal cutting-plane* heuristic based on ILP formula [18], for single target only, to find optimal solutions within practical computational limits. Their methods is a *branch-and-bound* approach that relaxes a large constraint set in order to find and improve the lower bound on the number of probes required in an optimal solution, until an optimal solution is obtained. The same data used in [18][23] were tested in their experiments. They demonstrated that their approach consistently found an optimal solution within 10 minutes, and was capable of reducing the number of probes required over the state-of-the-art heuristic methods by as much as 20%.

CHAPTER III

DETERMINISTIC GREEDY NON-UNIQUE PROBE SELECTION

III-1 Introduction

In this section we devise heuristics that filter out *bad probes* as in Meneses *et al.* [23]. In [23], Meneses *el al.* used no selection function to decide which probes to filter out; probes are removed as long as the feasibility of the given candidate solution is compromised. Also [23] used no random selection at any time in the algorithm. They initially sort the probes in increasing order of the number of targets they hybridize and then select probes, in this order, for inclusion in a candidate solution. The authors then scan this candidate probe set to test each probe for possible redundancy and remove any redundant probe. No additional information is used to direct the search. In the data sets, the range of the number of targets to which each probe hybridize is very small and many probes hybridize the same number of targets. Thus given two candidate probes, it is not easy to identify which probe is better than the other for inclusion into a candidate solution. In our methods, we propose some probe selection functions to guide the searching for optimal solution, so much more information about the probe set is stored in such a way that the algorithm can decide which probes to be selected for optimal solution.

In general, we want to select a minimum number of probes from the initial candidate probe set such that each target is c_{min}-coverd and each target-pair is s_{min}-

separated. Given a target probe incidence matrix H, the parameters c_{min} and s_{min}, the initial feasible candidate probe set P and the target set T, let P_{t_i} be the set of probes hybridizing to target t_i, and $P_{t_{ik}}$ be the set of probes separating the target-pair t_{ik}. It is clearly to see that there are m coverage (i.e., number of targets) and $\frac{m(m-1)}{2}$ separation (i.e., number of target-pairs). So we can define P_{min} as Equation 8:

$$P_{min} = \{ \bigcup_{1 \le i \le m} P_i \} \cup \{ \bigcup_{1 \le i \le k \le m} P_{ik} \} \tag{8}$$

where $P_i \subseteq P_{t_i}$ and $P_{ik} \subseteq P_{t_{ik}}$ are respectively coverage subsets and separation subsets selected for a minimal solution P_{min}.

A c_{min}-subset $P_i \subseteq P_{t_i}$ or a s_{min}-subset $P_{ik} \subseteq P_{t_{ik}}$ is an essential covering subset or separating subset, if and only if $P_i = P_{t_i}$ or $P_{ik} = P_{t_{ik}}$. In other words, if there are only c_{min} probes that hybridize to t_i or only s_{min} probes that separate t_{ik}, then those probes are *essential probes*. *Essential probes* must be contained in any minimal solution; that is, removing any such probe will make the solution infeasible. A *redundant probe* is the one for which a feasible solution remains feasibility when this probe is removed. Note that a probe may be redundant for some solutions but non-redundant for others. Thus there is a degree of redundancy between probes, with respect to minimal solutions. In this thesis, we assume that the initial candidate probe set is feasible. If not, we insert a sufficient number of unique virtual probes into P. For each target t_i or target-pair t_{ik} that a constraint is not satisfied, $(c_{min} - |P_{t_i}|)$ or $(s_{min} - |P_{t_{ik}}|)$ virtual unique probes are added.

III-2 Dominated Row Covering Heuristic (DRC)

III-2-1 Coverage Function

Given H, the parameter $c_{\min}$, the probe set $P = \{p_1, \ldots, p_n\}$ and the target set $T = \{t_1, \ldots, t_m\}$, we defined the function $\mathrm{cov}_{\mathrm{drc}} : P \times T \mapsto [0, 1]$ in [41] as follows:

$$\mathrm{cov}_{\mathrm{drc}}(p_j, t_i) = h_{ij} \times \frac{c_{\min}}{|P_{t_i}|}, \quad p_j \in P_{t_i}, \quad t_i \in T \tag{9}$$

where, P_{t_i} is the set of probes hybridizing to target t_i; $\mathrm{cov}_{\mathrm{drc}}(p_j, t_i)$ is the amount that p_j contributes to satisfy the coverage constraint for target t_i. For target t_i, p_j is likely to be redundant for a larger value of $|P_{t_i}|$ and likely to be non-redundant for a smaller value of $|P_{t_i}|$. We defined the *coverage function* $C_{\mathrm{drc}} : P \mapsto [0, 1]$ in [41] as follows:

$$C_{\mathrm{drc}}(p_j) = \max_{t_i \in T_{p_j}} \{\mathrm{cov}_{\mathrm{drc}}(p_j, t_i) \quad | \quad 1 \leq j \leq n\} \tag{10}$$

where T_{p_j} is the set of targets covered by p_j. $C_{\mathrm{drc}}(p_j)$ is the maximum amount that p_j can contribute to satisfy the minimum coverage constraints. Table 2 shows the coverage function table produced from Table 1. Function C_{drc} favors the selection

Table 2: Coverage function table obtained from Table 1 in DRC.

	p_1	p_2	p_3	p_4	p_5	p_6
t_1	$\frac{c_{\min}}{4}$	$\frac{c_{\min}}{4}$	0	$\frac{c_{\min}}{4}$	0	$\frac{c_{\min}}{4}$
t_2	$\frac{c_{\min}}{3}$	0	$\frac{c_{\min}}{3}$	0	0	$\frac{c_{\min}}{3}$
t_3	0	$\frac{c_{\min}}{5}$	$\frac{c_{\min}}{5}$	$\frac{c_{\min}}{5}$	$\frac{c_{\min}}{5}$	$\frac{c_{\min}}{5}$
t_4	0	0	$\frac{c_{\min}}{3}$	$\frac{c_{\min}}{3}$	$\frac{c_{\min}}{3}$	0
C_{drc}	$\frac{c_{\min}}{3}$	$\frac{c_{\min}}{4}$	$\frac{c_{\min}}{3}$	$\frac{c_{\min}}{3}$	$\frac{c_{\min}}{3}$	$\frac{c_{\min}}{3}$

of probes that $c_{\min}$-cover targets t_i that have the smallest subsets P_{t_i}; these are the essential or near-essential covering probes. In Table 2, for example, target t_2 has the minimal value $|P_{t_2}| = 3$, and hence any probe that covers it can be selected first. In particular, function C_{drc} guarantees the selection of near-essential covering probes that $c_{\min}$-cover *dominated targets*; t_i *dominates* t_k if $P_{t_k} \subset P_{t_i}$. In Table 2, for example, t_3 dominates t_4 since $P_{t_4} = \{p_3, p_4, p_5\} \subset \{p_2, p_3, p_4, p_5, p_6\} = P_{t_3}$. Any $c_{\min}$-cover of the dominated target t_k will also $c_{\min}$-cover all its dominant targets, and therefore, more targets are $c_{\min}$-covered. Probes covering the dominated target t_k have larger cov_{drc} values than probes covering its dominant targets t_i, since $|P_{t_k}| < |P_{t_i}|$, and hence they will be selected first.

We would also like to favor the selection of *dominant probes*; p_j *dominates* p_l if $T_{p_l} \subset T_{p_j}$. In Table 2, for instance, p_6 dominates p_1 since $T_{p_1} = \{t_1, t_2\} \subset \{t_1, t_2, t_3\} = T_{p_6}$. Selecting dominant probes instead of dominated probes covers more targets. In the example, however, we have $C_{\text{drc}}(p_1) = C_{\text{drc}}(p_6)$, and hence p_1 could be selected for target coverage rather than p_6, depending on a particular order of the probes. On the other hand, p_6 dominates p_2 and $C_{\text{drc}}(p_6) > C_{\text{drc}}(p_2)$, and hence p_6 will be selected first.

III-2-2 Separation Function

We want to choose the minimum number of probes such that each target-pair is $s_{\min}$-separated. We defined the function $\text{sep}_{\text{drc}} : P \times T^2 \mapsto [0, 1]$ as follows:

$$\text{sep}_{\text{drc}}(p_j, t_{ik}) = |h_{ij} - h_{kj}| \times \frac{s_{\min}}{|P_{t_{ik}}|}, \quad p_j \in P_{t_{ik}}, \quad t_{ik} \in T^2 \tag{11}$$

where, $P_{t_{ik}}$ is the set of probes separating target-pair t_{ik}; $\text{sep}_{\text{drc}}(p_j, t_{ik})$ is what p_j can contribute to satisfy the separation constraint for target-pair t_{ik}. We defined the *separation function* $S_{\text{drc}} : P \mapsto [0, 1]$ in [41] as follows:

$$S_{\text{drc}}(p_j) = \max_{t_{ik} \in T^2_{p_j}} \{\text{sep}_{\text{drc}}(p_j, t_{ik}) \quad | \quad 1 \leq j \leq n\} \tag{12}$$

where $T^2_{p_j}$ is the set of target-pairs separated by p_j. $S_{\text{drc}}(p_j)$ is the maximum amount that p_j can contribute to satisfy the minimum separation constraints. Table 3 shows the separation function table produced from Table 1. Function S_{drc} also favors the

Table 3: Separation function table obtained from Table 1 in DRC.

	p_1	p_2	p_3	p_4	p_5	p_6
t_{12}	0	$\frac{s_{\min}}{3}$	$\frac{s_{\min}}{3}$	$\frac{s_{\min}}{3}$	0	0
t_{13}	$\frac{s_{\min}}{3}$	0	$\frac{s_{\min}}{3}$	0	$\frac{s_{\min}}{3}$	0
t_{14}	$\frac{s_{\min}}{5}$	$\frac{s_{\min}}{5}$	$\frac{s_{\min}}{5}$	0	$\frac{s_{\min}}{5}$	$\frac{s_{\min}}{5}$
t_{23}	$\frac{s_{\min}}{4}$	$\frac{s_{\min}}{4}$	0	$\frac{s_{\min}}{4}$	$\frac{s_{\min}}{4}$	0
t_{24}	$\frac{s_{\min}}{4}$	0	0	$\frac{s_{\min}}{4}$	$\frac{s_{\min}}{4}$	$\frac{s_{\min}}{4}$
t_{34}	0	$\frac{s_{\min}}{2}$	0	0	0	$\frac{s_{\min}}{2}$
S_{drc}	$\frac{s_{\min}}{3}$	$\frac{s_{\min}}{2}$	$\frac{s_{\min}}{3}$	$\frac{s_{\min}}{3}$	$\frac{s_{\min}}{3}$	$\frac{s_{\min}}{2}$

selection of probes that $s_{\min}$-separate target-pairs t_{ik} which have the smallest subsets $P_{t_{ik}}$ and further favors the selection of near-essential separating probes that $s_{\min}$-separate *dominated target pairs.*

III-2-3 Selection Function

We want to select the minimum number of probes such that all coverage and separation constraints are satisfied; that is, we must select a probe according to its ability to help satisfy both coverage *and* separation constraints. We combined functions C_{drc} and S_{drc} into a single probe selection function, $D_{\text{drc}} : P \mapsto [0, 1]$ as follows:

$$D_{\text{drc}}(p_j) = \max\{(C_{\text{drc}}(p_j), S_{\text{drc}}(p_j)) \mid 1 \leq j \leq n\} \tag{13}$$

$D_{\text{drc}}(p_j)$ is the degree of contribution of p_j, that is, the maximum amount required for p_j to satisfy all constraints. D_{drc} ensures that all essential probes p_j will be selected for inclusion in the subsequent candidate solution, since $C_{\text{drc}}(p_j) = 1$ or $S_{\text{drc}}(p_j) = 1$. With our definition of D_{drc}, probes p that cover dominated targets or separate dominated target-pairs have the highest $D_{\text{drc}}(p)$ values.

III-2-4 Algorithm

Our heuristic consists of three phases: *Initialization Phase*, *Construction Phase*, and *Reduction Phase*. In the *Initialization Phase*, we compute the initial $D(p)$ value for each probe $p \in P$ given matrix H and create an initial and possibly non-feasible solution P_{ini} containing essential probes only. In the *Construction Phase*, we repeatedly insert high-degree probes into P_{ini} until an initial feasible solution P_{sol} is obtained. In the *Reduction Phase*, we reduce P_{sol} by repeatedly removing low-degree probes such as to obtain a final near minimal feasible solution P_{min}.

ALGORITHM 1 Dominated Row Covering Heuristic (DRC)

Input: $T = \{t_1, \ldots, t_m\}$, $P = \{p_1, \ldots, p_n\}$, and $H = [h_{ij}]$
Output: Near-minimal solution $P_{\min}$

{Initialization Phase}
Compute $D_{\mathrm{drc}}(p)$ for all $p \in P$ using Equations 9-13
$P_{ini} \leftarrow \{p \in P | D(p) = 1\}$ {essential probes}
{Construction Phase}
$P_{sol} \leftarrow P_{ini}$
Sort $P \setminus P_{sol}$ in decreasing order of $D(p)$
for each target t_i not c_{min} − *covered* by P_{sol} **do**
  $n_i \leftarrow$ # probes needed to complete c_{min} − *covered* of t_i
  $P_{sol} \leftarrow P_{sol} \cup \bigcup_1^{n_i}\{$ next highest degree probe $p_l \in P \setminus P_{sol}$ that covers $t_i\}$
end for
for each target pair t_{ik} not s_{min}-separated by P_{sol} **do**
  $n_{ik} \leftarrow$ # probes needed to complete s_{min}-separation of t_{ik}
  $P_{sol} \leftarrow P_{sol} \cup \bigcup_1^{n_{ik}}\{$ next highest degree probe $p_l \in P \setminus P_{sol}$ that separates t_{ik}
end for
{Reduction Phase}
$P_{min} \leftarrow P_{sol}$
$H \leftarrow H|P_{min}$ {update H to probes in P_{min}}
Compute $D(p)$ for all $p \in P_{min}$
Sort $P_{del} \leftarrow \{p \in P_{min} | D(p) < 1\}$ in increasing order
if $P_{min} \setminus \{p\}$ is feasible for each $p \in P_{del}$ **then**
  $P_{min} \leftarrow P_{min} \setminus \{p\}$
end if
Return $P_{\min}$

III-2-5 Computational Complexity

In heuristic DRC, the computational complexity for calculation of coverage function is $O(mn)$; $O(m^2n)$ for calculation of separation function, so the computational complexity for selection function $D_{\text{drc}}(p)$ is $O(m^2n)$. For the *Construction Phase*, the complexity for sorting $P \setminus P_{sol}$ is $O(n \log n)$; the complexity for coverage-construction is $O(mn)$ for the worst case; $O(m^2n)$ for separation-construction in the worst case. While for the *Reduction Phase*, the computational complexity is $O(m^2n + n \log n)$. So finally, the computational complexity for heuristic DRC is $O(m^2n + n \log n)$.

III-3 Dominated Probe Selection Heuristic (DPS)

III-3-1 Coverage Function

To favor the selection of a dominant probe among dominated probes equal in value C_{drc}, we penalize each probe p by an amount proportional to $|T_p|$, as follows:

$$C_{\text{dps}}(p_j) = C_{\text{drc}}(p_j) \times \frac{1}{m - |T_{p_j}| + 1} \tag{14}$$

and probes that cover fewer targets are penalized more than probes that cover more targets. Note: here $|T_{p_j}| < m$ is always true, because the probe that hybridizes with all targets is useless for the design, and can not be selected in the candidate probe pool. Table 4 shows the values of C_{dps} for each probe.

Table 4: Coverage function table obtained from Table 1 in DPS.

	p_1	p_2	p_3	p_4	p_5	p_6
t_1	$\frac{c_{\min}}{12}$	$\frac{c_{\min}}{12}$	0	$\frac{c_{\min}}{8}$	0	$\frac{c_{\min}}{8}$
t_2	$\frac{c_{\min}}{9}$	0	$\frac{c_{\min}}{6}$	0	0	$\frac{c_{\min}}{6}$
t_3	0	$\frac{c_{\min}}{15}$	$\frac{c_{\min}}{10}$	$\frac{c_{\min}}{10}$	$\frac{c_{\min}}{15}$	$\frac{c_{\min}}{10}$
t_4	0	0	$\frac{c_{\min}}{6}$	$\frac{c_{\min}}{6}$	$\frac{c_{\min}}{9}$	0
C_{dps}	$\frac{c_{\min}}{9}$	$\frac{c_{\min}}{12}$	$\frac{c_{\min}}{6}$	$\frac{c_{\min}}{6}$	$\frac{c_{\min}}{9}$	$\frac{c_{\min}}{6}$

III-3-2 Separation Function

To favor the selection of a dominant probe that has the same value, S_{drc}, as some of its dominated probes, we penalize each probe p by an amount proportional to $|T_p^2|$, as follows:

$$S_{\text{dps}}(p_j) = S_{\text{drc}}(p_j) \times \frac{1}{\frac{m(m-1)}{2} - |T_{p_j}^2| + 1} \tag{15}$$

and probes that separate fewer target-pairs are penalized more than probes that separate more target-pairs. Note: $\frac{m(m-1)}{2} > |T_{p_j}^2|$ is also always true, when $m > 2$.

III-3-3 Selection Function

In this paper, we use the following probe selection function, $D_{\text{dps}} : P \mapsto [0, 1]$:

$$D_{\text{dps}}(p_j) = \max\{(C_{\text{dps}}(p_j), S_{\text{dps}}(p_j)) \mid 1 \leq j \leq n\} \tag{16}$$

to favor the dominant probes among all probes that have equal values in D_{drc}; this is the secondary greedy selection principle. These two greedy principles together allow larger coverage and separation when using D_{dps} than D_{drc} in a greedy search method.

III-3-4 Algorithm

The Dominant Probe Selection (DPS) heuristic, is similar to DRC in Section III-2 except the definition of $D(p)$, so the algorithm of DPS is almost same as that in Section III-2-4 except the calculation of $D(p)$.

III-3-5 Computational Complexity

Heuristic DPS also performs similar with DRC except the calculation of selection function $D_{\text{dps}}(p)$. While we use two stacks with length n to store $|T_p|$ and $|T_p^2|$ respectively, so the computational complexity for the calculation of selection function is still $O(m^2n)$ in the worst case. Then the computational complexity for heuristic DPS is also $O(m^2n + n\log n)$.

III-4 Normalized Dominant Probe Selection Heuristic (DPSn)

III-4-1 Coverage Function

Compared with DRC and DPS, The difference of DPSn is that we normalized the contribution of each target to c_{min} as following:

$$\text{cov}_{\text{dpsn}}(p_j, t_i) = \gamma_i \times h_{ij} \times \frac{1}{m - |T_{p_j}| + 1} \tag{17}$$

where, P_{t_i} is the set of probes hybridizing to target t_i; $\text{cov}_{\text{drc}}(p_j, t_i)$ is the amount that p_j contributes to satisfy the coverage constraint for target t_i. As explained in Section III-3, $|T_{p_j}| < m$ is always true. The normalization factor γ_i is given below:

$$\gamma_i = \frac{c_{min}}{\sum_{j=1}^{j=n} \frac{h_{ij}}{m-|T_{p_j}|+1}} \tag{18}$$

We defined the *coverage function* C_{dpsn}:

$$C_{\text{dpsn}}(p_j) = \max_{t_i \in T_{p_j}} \{\text{cov}_{\text{dpsn}}(p_j, t_i) \quad | \quad 1 \leq j \leq n\} \tag{19}$$

where T_{p_j} is the set of targets covered by p_j.

III-4-2 Separation Function

Similarly, we normalized the contribution of each target pair to s_{min} in Equation 20.

$$\text{sep}_{\text{dpsn}}(p_j, t_{ik}) = \sigma_{ik} \times |h_{ij} - h_{kj}| \times \frac{1}{\frac{m(m-1)}{2} - |T^2_{p_j}| + 1} \tag{20}$$

where, $P_{t_{ik}}$ is the set of probes separating target-pair t_{ik}. The normalization factors σ_{ik} are given below:

$$\sigma_{ik} = \frac{s_{min}}{\sum_{j=1}^{j=n} \frac{|h_{ij}-h_{kj}|}{\frac{m(m-1)}{2}-|T^2_{p_j}|+1}} \tag{21}$$

$$S_{\text{dpsn}}(p_j) = \max_{t_{ik} \in T^2_{p_j}} \{\text{sep}_{\text{dpsn}}(p_j, t_{ik}) \quad | \quad 1 \leq j \leq n\} \tag{22}$$

where $T^2_{p_j}$ is the set of target-pairs separated by p_j.

III-4-3 Selection Function

We use similar selection function as in DRC and DPS.

$$D_{\text{dpsn}}(p_j) = \max\{(C_{\text{dpsn}}(p_j), S_{\text{dpsn}}(p_j)) \mid 1 \leq j \leq n\} \tag{23}$$

III-4-4 Algorithm

The algorithm in DPSn is also almost same as that in SectionIII-2-4, except the calculation of $D(p)$.

III-4-5 Computational Complexity

Although, heuristic DPSn implements more complicate selection function than DRC and DPS, the complexity still keep same in the worst case for the calculation of selection function. So the the computational complexity for DPSn is still $O(m^2n + n\log n)$.

III-5 Dynamic DRC, DPS and DPSn Heuristics

In DRC and DPS, given the target-probe incidence matrix H, the entries in the coverage matrix (Table 2) and the separation matrix (Table 3) are computed in the *Initialization Phase* and remain un-changed during the *Construction Phase* until the *Reduction Phase* where we compute a new incidence matrix $H = H|_{P_{\min}}$. The *next probe* is selected without considering the current set of probes that are already se-

lected, nor, the current set of rows (targets and target-pairs) that are already covered by the current candidate probe set.

The *Dynamic Dominated Row Covering Heuristic* (DDRC), *Dynamic Dominant Probe Selection Heuristic* (DDPS) and *Normalized Dynamic Dominant Probe Selection* (DDPSn) make use of knowledge that can help achieve greater reduction: 1) which probes are already selected, 2) which rows are covered by already selected probes and 3) how many more probes are needed to satisfy the constraints for each row. For example, if we remove the already selected probes (that is, if we remove from H the columns associated to already selected probes) and update for each row the number of remaining probes required to $c_{\min}$-cover or $s_{\min}$-cover that row, then some dominant row may become dominated and therefore the algorithm can concentrate its efforts to select probes for covering this new dominated row along with the current dominated rows. Likewise, once a row is already $c_{\min}$-covered or $s_{\min}$-covered the algorithm should concentrate its efforts on selecting probes for the remaining rows only. DDRC, DDPS and DDPSn are *dynamic* in the sense that entries $\mathrm{cov}(p_j, t_i)$ and $\mathrm{sep}(p_j, t_{ik})$ are updated only for rows t_i and t_{ik} covered by the newly selected probe $p_{l \neq j}$, each time a new probe is selected. In DDRC, we first initialize the solution with *essential* probes. Then let p_l be the newly selected *non-essential* probe, in DDRC, we update the cov and sep values only in those rows t_i and t_{ik} that are covered by p_l as

$$\mathrm{cov}(p_{j \neq l}, t_i) = h_{ij} \times \frac{c_{\min} - |C_{t_i}|}{|P_{t_i}| - |C_{t_i}|}, \quad p_j \in P_{t_i} \setminus C_{t_i}, \quad t_i \in T \tag{24}$$

$$\text{sep}(p_{j\neq l}, t_{ik}) = |h_{ij} - h_{kj}| \times \frac{s_{\min} - |S_{t_{ik}}|}{|P_{t_{ik}}| - |S_{t_{ik}}|}, \quad p_j \in P_{t_{ik}} \setminus S_{t_{ik}}, \quad t_{ik} \in T^2 \tag{25}$$

where C_{t_i} and $S_{t_{ik}}$ are respectively, the set of selected probes (including new selected probe p_l) that already cover rows t_i and t_{ik}. Note: here p_l and p_j are *non-essential* probes, so $|P_{t_i}| > c_{min}$ and $|P_{t_{ik}}| > s_{min}$ are always true. For target or target-pair that has not been c_{min} covered or s_{min} separated, $|P_{t_i}| > c_{min} \geq |C_{t_i}|$ and $|P_{t_{ik}}| > s_{min} \geq |S_{t_{ik}}|$. We then update the matrix H as

$$H = H|_{P \setminus \{p_l\}} \tag{26}$$

or simply set $h_{il} = 0$ for $1 \leq i \leq m$. Table 5 shows one example, where if p_1 is selected, given the left coverage matrix, then p_1 is removed in the right coverage matrix

Table 5: Coverage matrix for DDRC before and after selectiong p_1.

	p_1	p_2	p_3	p_4	p_5	p_6
t_1	$\frac{c_{\min}}{4}$	$\frac{c_{\min}}{4}$	0	$\frac{c_{\min}}{4}$	0	$\frac{c_{\min}}{4}$
t_2	$\frac{c_{\min}}{3}$	0	$\frac{c_{\min}}{3}$	0	0	$\frac{c_{\min}}{3}$
t_3	0	$\frac{c_{\min}}{5}$	$\frac{c_{\min}}{5}$	$\frac{c_{\min}}{5}$	$\frac{c_{\min}}{5}$	$\frac{c_{\min}}{5}$
t_4	0	0	$\frac{c_{\min}}{3}$	$\frac{c_{\min}}{3}$	$\frac{c_{\min}}{3}$	0
C_{drc}	$\frac{c_{\min}}{3}$	$\frac{c_{\min}}{4}$	$\frac{c_{\min}}{3}$	$\frac{c_{\min}}{3}$	$\frac{c_{\min}}{3}$	$\frac{c_{\min}}{3}$

$\Rightarrow$

	p_1	p_2	p_3	p_4	p_5	p_6
t_1		$\frac{c_{\min}-1}{3}$	0	$\frac{c_{\min}-1}{3}$	0	$\frac{c_{\min}-1}{3}$
t_2		0	$\frac{c_{\min}-1}{2}$	0	0	$\frac{c_{\min}-1}{2}$
t_3		$\frac{c_{\min}}{5}$	$\frac{c_{\min}}{5}$	$\frac{c_{\min}}{5}$	$\frac{c_{\min}}{5}$	$\frac{c_{\min}}{5}$
t_4		0	$\frac{c_{\min}}{3}$	$\frac{c_{\min}}{3}$	$\frac{c_{\min}}{3}$	0
C_{drc}		-	-	-	$\frac{c_{\min}}{3}$	-

The DDPS is similar to DDRC except that functions C, S and D are defined using Equations (27) and (28) below.

$$\text{cov}(p_{j\neq l}, t_i) = h_{ij} \times \frac{c_{\min} - |C_{t_i}|}{|P_{t_i}| - |C_{t_i}|} \times \frac{1}{m - (|T_{p_j}| - |U_{p_j}|)} \tag{27}$$

$$\text{sep}(p_{j\neq l}, t_{ik}) = |h_{ij} - h_{kj}| \times \frac{s_{\min} - |S_{t_{ik}}|}{|P_{t_{ik}}| - |S_{t_{ik}}|} \times \frac{1}{\frac{m(m-1)}{2} - (|T^2_{p_j}| - |U^2_{p_j}|)} \tag{28}$$

where $U_{p_j} \subseteq T_{p_j}$ and $U^2_{p_j} \subseteq T^2_{p_j}$ are, respectively, the set of targets in T_{p_j} and target-pairs in $T^2_{p_j}$ that are already $c_{\min}$-covered and $s_{\min}$-separated by the currently selected probe set. As explained before, the rows associated with these targets or target-pairs will be all-zero, and therefore, they should be discarded from T_{p_j} or $T^2_{p_j}$ for given probe p_j.

In DDPSn, we use Equation 29-32 to normalize only those targets and target-pairs affected by the selection of p_l to $c_{min} - |C_{t_i}|$ and $|s_{min} - S_{t_{ik}}|$ respectively.

$$\text{cov}(p_{j\neq l}, t_i) = \gamma_i \times h_{ij} \times \frac{1}{m - (|T_{p_j}| - |U_{p_j}|)} \tag{29}$$

and

$$\text{sep}(p_{j\neq l}, t_{ik}) = \sigma_{ik} \times |h_{ij} - h_{kj}| \times \frac{1}{\frac{m(m-1)}{2} - (|T^2_{p_j}| - |U^2_{p_j}|)} \tag{30}$$

where $U_{p_j} \subseteq T_{p_j}$ and $U^2_{p_j} \subseteq T^2_{p_j}$ are, respectively, the set of targets in T_{p_j} and target-pairs in $T^2_{p_j}$ that are already $c_{\min}$-covered and $s_{\min}$-separated by the currently selected probe set. The normalization factors γ_i and σ_{ik} are given below:

$$\gamma_i = \frac{c_{min} - |C_{t_i}|}{\sum_{j\neq l} \frac{h_{ij}}{m-(|T_{p_j}|-|U_{p_j}|)}} \tag{31}$$

$$\sigma_{ik} = \frac{s_{min} - |S_{t_{ik}}|}{\sum_{j\neq l} \frac{|h_{ij}-h_{kj}|}{\frac{m(m-1)}{2}-(|T^2_{p_j}|-|U^2_{p_j}|)}} \tag{32}$$

III-5-1 Algorithms

The algorithm presented in DDRC is described as Algorithm 2.

III-5-2 Computational Complexity

In dynamic heuristics, we update selection function values $D(p)$ once add one *non-essential* probe into solution. Because, in DDRC and DDPS, we just update the *cov* and *sep* values only in those rows t_i and t_{ik} that are covered by p_l, which is the newly selected *non-essential* probe, so the computational complexity for updating is $O(tq)$, where $t = \max\{(|T_{p_j}|, |T^2_{p_j}|) \mid 1 \leq j \leq n\}$ and q is the number of current unselected candidate probes. But in DDPSn, we have to update normalization factors γ_i and σ_{ik} for all *unselected* probes with computational complexity $O(m^2q)$, where q is the number of current unselected candidate probes. While, the updating occurs at most $n-1$ times in the worst case, when all candidate probes are included in the final solution. So we can see that the computational complexity for dynamic heuristics DDRC and DDPS is $O(m^2n + tn^2)$, where $t = \max\{(|T_{p_j}|, |T^2_{p_j}|) \mid 1 \leq j \leq n\}$; $O(m^2n^2)$ is the computational complexity in the worst case for dynamic heuristic DDPSn.

ALGORITHM 2 Dynamic Dominated Row Covering Heuristic (DDRC)

Input: $T = \{t_1, \ldots, t_m\}$, $P = \{p_1, \ldots, p_n\}$, and $H = [h_{ij}]$
Output: Near-minimal solution $P_{\min}$

{Initialization Phase}
$G \leftarrow H$
$P_{ini} \leftarrow \{p \in P | p \text{ is essential } \}$
for all $t_a (1 \leq a \leq m)$ and $t_{ab} (1 \leq a < b \leq m)$ covered by each $q \in P_{ini}$ **do**
  Compute $D(p)$ for all $p \in \{P_{t_a} \setminus C_{t_a}\} \cup \{P_{t_{ab}} \setminus S_{t_{ab}}\}$
end for
$H \leftarrow H|P \setminus P_{ini}$ {update H to probes in $P \setminus P_{ini}$}
$P \leftarrow P \setminus P_{ini}$
{Construction Phase}
$P_{sol} \leftarrow P_{ini}$
for each target t_i not c_{min} covered by P_{sol} **do**
  $n_i \leftarrow$ # probes needed to complete c_{min}-coverage of t_i
  repeat
    $P_{sol} \leftarrow P_{sol} \cup \{q \in P \setminus P_{sol}$ with highest degree that covers $t_i\}$
    for all $t_a (1 \leq a \leq m)$ and $t_{ab} (1 \leq a < b \leq m)$ covered by q **do**
      Update $D(p)$ for all $p \in \{P_{t_a} \setminus C_{t_a}\} \cup \{P_{t_{ab}} \setminus S_{t_{ab}}\}$
    end for
    $H \leftarrow H|P \setminus \{q\}$
    $P \leftarrow P \setminus \{q\}$
  until n_i probes are inserted
end for
for each target pair t_{ik} not s_{min} separated by P_{sol} **do**
  $n_{ik} \leftarrow$ # probes needed to complete s_{min} separation of t_{ik}
  repeat
    $P_{sol} \leftarrow P_{sol} \cup \{$ probe $q \in P \setminus P_{sol}$ with highest degree that separate $t_{ik}\}$
    for all $t_a (1 \leq a \leq m)$ and $t_{ab} (1 \leq a < b \leq m)$ covered by q **do**
      Update $D(p)$ for all $p \in \{P_{t_a} \setminus C_{t_a}\} \cup \{P_{t_{ab}} \setminus S_{t_{ab}}\}$
    end for
    $H \leftarrow H|P \setminus \{q\}$
    $P \leftarrow P \setminus \{q\}$
  until n_{ik} probes are inserted
end for
{Reduction Phase}
$P_{min} \leftarrow P_{sol}$
$H \leftarrow G|P_{min}$ {we restore initial H and restrict to P_{min}}
Compute $D(p) = D_{drc}(p)$ for all $p \in P_{min}$
Sort $P_{del} \leftarrow \{p \in P_{min} | D(p) < 1\}$ in increasing order
if $P_{min} \setminus \{p\}$ is feasible for each $p \in P_{del}$ **then**
  $P_{min} \leftarrow P_{min} \setminus \{p\}$
end if
Return final P_{min}

III-6 Sequential Forward Probe Selection Algorithm (SFPS)

In this section, a sub-optimal technique from pattern recognition is applied for the first time, to the non-unique probe selection problem. In particular, the well-known *sequential forward selection* (SFS) algorithm [24], for feature subset selection, is adapted to find near-minimal feasible probe sets [45]. Feature selection (FS) constitutes one of the two principal phases of pattern recognition system design, the other being the design of pattern classification stage which employs the selected features. The main goal of FS is to select a subset of d features from the given set of D measurements, $d < D$, without significantly degrading (or, with possibly improving) the performance of the recognition system. Given a suitable criterion function for assessing the *effectiveness* of feature subsets to classify data, FS is reduced to a combinatorial search problem that finds an optimal subset based on the selected measure.

A microarray design experiment is a pattern recognition system where the measurements are provided by a biological sample and a target set (augmented with the set of all target-pairs, if non-unique probes are used), and where the classifier system is a probe set that classifies each target, or target-pair, as present or absent in the sample. However, with microarrays, the problem is to reduce the complexity of the classifier system (i.e., the size of the probe set) while still able to correctly classify each target and target-pair as present or absent in the biological sample. Here, the feature space representing the sample, which includes the targets and the target-pairs, is not subject to optimization.

We adapt the SFS to find a near minimal probe set as follows: the best probe set is constructed by adding, to the current non-feasible probe set, one probe at a time until we obtain a feasible probe set with the hope it has the least cardinality u. More specifically, to form the best feasible subset of probes, the starting point of the search is the empty set, $P^{1\ldots0}$, which is then successively built up. This is known as the bottom up approach. This method is generally sub-optimal since the best probe is always added to a working subset of probes, $P^{1\ldots u}$.

III-6-1 Subset Selection Criteria

In this section, we define the criteria required to decide which is the best subset to select. Let $P^{1\ldots u} = \{q_1, \ldots, q_u\} \subseteq P$ be a probe set to be evaluated, where $q_j \in P$, $1 \leq j \leq u$ and $1 \leq u \leq n$, and $P^{1\ldots0} = \emptyset$. $P^{1\ldots u}$ $c_{\min}$-covers a target t_i if at least $c_{\min}$ probes in $P^{1\ldots u}$ cover t_i. $P^{1\ldots u}$ $s_{\min}$-separates a target-pair t_{ik} if at least $s_{\min}$ probes in $P^{1\ldots u}$ separate t_{ik}. Our aim is to select the subset $P^{1\ldots u}$ which $c_{\min}$-covers as many target as possible and $s_{\min}$-separates as many target-pairs as possible, or, which satisfies all the constraints with the least cardinality u.

III-6-1-1 Coverage Criterion

Given a collection $\mathcal{P} \subseteq 2^P$, we want to choose the subset $P^{1\ldots u} \subseteq P$ such that each target is $c_{\min}$-covered by $P^{1\ldots u}$. Given the matrix H, the parameter $c_{\min}$, the candidate probe set $P = \{p_1, \ldots, p_n\}$ and the target set $T = \{t_1, \ldots, t_m\}$; to evaluate the ability of subset $P^{1\ldots u}$ to $c_{\min}$-cover T, we generalize the coverage function as follows:

$$C_{\text{dps}}(P^{1\ldots u}) = \max_{t_i \in T_{P1\ldots u}} \left\{ \sum_{j=1}^{j=u} \text{cov}_{\text{dps}}(q_j, t_i) \quad | \quad q_j \in P^{1\ldots u} \right\} \tag{33}$$

where $T_{P^{1\ldots u}} = T_{q_1} \cup \ldots \cup T_{q_u}$ is the set of targets covered by $P^{1\ldots u}$. $C_{\text{dps}}(P^{1\ldots u}) : 2^P \mapsto \Re^+$ is the maximum amount that $P^{1\ldots u}$ can contribute to satisfy the minimum coverage constraints. Table 6 shows an example of a subset coverage table obtained from Table 1, given five subsets. In the example, P_{ab} means the subset $\{q_a, q_b\}$. We also show, for P_{31}, the computation of Equation (33). Clearly, $C_{\text{dps}}(P^{1\ldots u})$ is maximal

Table 6: Example of subset coverage obtained from Table 1.

	$\{p_3\}$	$\cup$	$\{p_1\}$	$=$	P_{31}	P_{32}	P_{34}	P_{35}	P_{36}
t_1	0	+	$\frac{c_{\min}}{4}\frac{1}{3}$	=	$\frac{c_{\min}}{12}$	$\frac{c_{\min}}{12}$	$\frac{c_{\min}}{8}$	0	$\frac{c_{\min}}{8}$
t_2	$\frac{c_{\min}}{3}\frac{1}{2}$	+	$\frac{c_{\min}}{3}\frac{1}{3}$	=	$\frac{5c_{\min}}{18}$	$\frac{c_{\min}}{6}$	$\frac{c_{\min}}{6}$	$\frac{c_{\min}}{6}$	$\frac{c_{\min}}{3}$
t_3	$\frac{c_{\min}}{5}\frac{1}{2}$	+	0	=	$\frac{c_{\min}}{10}$	$\frac{c_{\min}}{6}$	$\frac{c_{\min}}{5}$	$\frac{c_{\min}}{6}$	$\frac{c_{\min}}{5}$
t_4	$\frac{c_{\min}}{3}\frac{1}{2}$	+	0	=	$\frac{c_{\min}}{6}$	$\frac{c_{\min}}{6}$	$\frac{c_{\min}}{3}$	$\frac{5c_{\min}}{18}$	$\frac{c_{\min}}{6}$
C_{dps}					$\frac{5c_{\min}}{18}$	$\frac{c_{\min}}{6}$	$\frac{c_{\min}}{3}$	$\frac{5c_{\min}}{18}$	$\frac{c_{\min}}{3}$

if $C_{\text{dps}}(q_j)$ is maximal for each $q_j \in P^{1\ldots u}$. Thus, for subsets of probes, function C_{dps} favors the selection of those subsets that contain probes having the highest coverage values. For example in Table 6, probes p_3, p_4 and p_6 have the highest coverage values (shown in Table 4), and hence, subsets such as P_{34} and P_{36} have the best values. C_{dps} indicates only how much a subset contributes in satisfying the coverage constraints, not how well the subset satisfies the coverage constraints. For instance, in the table, subsets P_{31} and P_{35} produce a tie, but P_{31} should be preferred since it covers more targets. Also, between the two subsets, which attain the same value of C_{dps}, the one that satisfies all coverage constraints (or, closer to satisfying all coverage constraints) should be preferred. We define the *coverage criterion*, $F_{C_{\text{dps}}} : 2^P \mapsto \Re^+$, as follows:

$$F_{C_{\text{dps}}}(P^{1\ldots u}) = C_{\text{dps}}(P^{1\ldots u}) \times \frac{|T_{P^{1\ldots u}}| - |U_{P^{1\ldots u}}|}{m - |U_{P^{1\ldots u}}|} \times \frac{\sum_{t_i \in T \setminus U_{P^{1\ldots u}}} \text{fea}\left(P_{t_i}^{1\ldots u}\right)}{(m - |U_{P^{1\ldots u}}|) \cdot c_{\min}} \tag{34}$$

where, $U_{P^{1\ldots u}}$ is the set of targets already $c_{\min}$-covered by $P^{1\ldots u}$ (probes need not be selected to cover such targets); $P_{t_i}^{1\ldots u}$ is the set of probes in $P^{1\ldots u}$ that cover t_i, and $\text{fea} : 2^P \mapsto \Re^+$ defined as

$$\text{fea}\left(P_{t_i}^{1\ldots u}\right) = \begin{cases} \left|P_{t_i}^{1\ldots u}\right| & , \text{ if } \left|P_{t_i}^{1\ldots u}\right| < c_{\min} \\ c_{\min} & , \text{ otherwise} \end{cases} \tag{35}$$

specifies how much the coverage constraint is satisfied on t_i; the sum equals $(m - |U_{P^{1\ldots u}}|)\, c_{\min}$ when all coverage constraints are satisfied. Hence, the second term penalizes subsets that cover fewer targets and the third term penalizes subsets that satisfy fewer coverage constraints. $F_{C_{\text{dps}}}$ is maximal when all three terms are maximal.

III-6-1-2 Separation Criterion

The derivation of the *separation criterion* is similar to that of coverage, except that we use terms and variables related to separation; such as, target-pair, $s_{\min}$, and so on, in the equations below. Given a collection $\mathcal{P} \subseteq 2^P$, we want to choose the subset $P^{1\ldots u} \subseteq P$ such that each target-pair is $s_{\min}$-separated by $P^{1\ldots u}$. Consider the matrix H, the parameter $s_{\min}$, the candidate probe set $P = \{p_1, \ldots, p_n\}$ and the target set $T = \{t_1, \ldots, t_m\}$. Following the same reasoning as in Section III-6-1-1, we obtain the following equations for separation:

$$S_{\text{dps}}(P^{1\ldots u}) = \max_{t_{ik} \in T^2_{P1\ldots u}} \left\{ \sum_{j=1}^{j=u} \text{sep}_{\text{dps}}(q_j, t_{ik}) \quad | \, q_j \in P^{1\ldots u} \right\} \tag{36}$$

where $T^2_{P1\ldots u} = T^2_{q_1} \cup \ldots \cup T^2_{q_u}$ is the set of target-pairs separated by $P^{1\ldots u}$. $S_{\text{dps}}(P^{1\ldots u}) : 2^P \mapsto \Re^+$ is the maximum amount that $P^{1\ldots u}$ can contribute to satisfy the minimum separation constraints. The *separation criterion* is given by:

$$F_{S_{\text{dps}}}(P^{1\ldots u}) = S_{\text{dps}}(P^{1\ldots u}) \times \frac{\left|T^2_{P1\ldots u}\right| - \left|U^2_{P1\ldots u}\right|}{\frac{m(m-1)}{2} - \left|U^2_{P1\ldots u}\right|} \times \frac{\sum_{t_{ik} \in T^2 \setminus U^2_{P1\ldots u}} \text{fea}\left(P^{1\ldots u}_{t_{ik}}\right)}{\left(\frac{m(m-1)}{2} - \left|U^2_{P1\ldots u}\right|\right) \cdot s_{\min}} \tag{37}$$

where, $U^2_{P1\ldots u}$ is the set of target-pairs already $s_{\min}$-separated by $P^{1\ldots u}$ (probes need not be selected to separate such target-pairs); $P^{1\ldots u}_{t_{ik}}$ is the set of probes in $P^{1\ldots u}$ that separate t_{ik}, and $\text{fea} : 2^P \mapsto \Re^+$ defined as

$$\text{fea}\left(P^{1\ldots u}_{t_{ik}}\right) = \begin{cases} \left|P^{1\ldots u}_{t_{ik}}\right| & , \text{ if } \left|P^{1\ldots u}_{t_{ik}}\right| < s_{\min} \\ s_{\min} & , \text{ otherwise} \end{cases} \tag{38}$$

specifies how much the separation constraint is satisfied on t_{ik}; the sum equals $\left(\frac{m(m-1)}{2} - \left|U^2_{P1\ldots u}\right|\right) s_{\min}$ when all separation constraints are satisfied. Thus, the second term penalizes subsets that separate fewer target-pairs and the third term penalizes subsets that satisfy fewer separation constraints. $F_{S_{\text{dps}}}$ is maximal when all three terms are maximal.

III-6-1-3 Selection Criterion

We combine both the coverage criterion and the separation criterion into a single subset *selection criterion*

$$F_{D_{\text{dps}}}(P^{1\ldots u}) = \max\left\{ F_{C_{\text{dps}}}(P^{1\ldots u}) \, , \, F_{S_{\text{dps}}}(P^{1\ldots u}) \right\} \tag{39}$$

which specifies the degree to which a subset of probes satisfies *all* constraints.

III-6-2 Algorithms

The *sequential forward probe selection* (SFPS) method (Algorithm 3) is based on the SFS algorithm. SFPS uses the $F_{D_{\text{dps}}}$ function as the criterion for selecting the best subset among a collection of probe sets. The best probe, q^+, to insert in a working subset, $P^{1\ldots u}$, is the one that maximizes the criterion, $F_{D_{\text{dps}}}$, when it is included. SFPS terminates when $P^{1\ldots u}$ is feasible; which is then reduced to a near-minimal solution, P_{min}, in Algorithm 4, by removing the redundant probes. SFPS

ALGORITHM 3 Sequential Forward Probe Selection (SFPS)

Input: $T = \{t_1, \ldots, t_m\}$, $P = \{p_1, \ldots, p_n\}$, and $H = [h_{ij}]$
Output: Near-minimal solution P_{min}

1: Compute $D_{\text{dps}}(p)$ for all $p \in P$
2: $u \leftarrow$ number of essential probes
3: $P^{1\ldots u} \leftarrow$ set of essential probes
4: **repeat**
5: $\quad q^+ \leftarrow \arg\max_{q \in P \setminus P^{1\ldots u}} F_{D_{\text{dps}}}\left(P^{1\ldots u} \cup \{q\}\right)$
6: $\quad P^{1\ldots(u+1)} \leftarrow P^{1\ldots u} \cup \{q^+\}$
7: $\quad u \leftarrow u + 1$
8: **until** $P^{1\ldots u}$ is feasible
9: Return $P_{\text{min}} \leftarrow$ Reduction$(P^{1\ldots u}, P, T, H)$

locally searches the power set, 2^P, of the probe set P. That is, at each subset selection step, the neighborhood of the working subset $P^{1\ldots u} \in 2^P$ is the collection $\mathcal{P}^{1\ldots(u+1)} = \{P^{1\ldots u} \cup \{q_1\}, P^{1\ldots u} \cup \{q_2\}, \ldots, P^{1\ldots u} \cup \{q_{n-u}\}\} \subset 2^P$, $q_j \in P \setminus P^{1\ldots u}$ for $1 \leq j \leq n-u$. The subset to select is the one in $\mathcal{P}^{1\ldots(u+1)}$ that maximizes the criterion

ALGORITHM 4 Reduction in SFPS

Input: $P^{1\ldots u}$, P, T, H
Output: Reduced solution P_{red}

1: $P_{\text{red}} \leftarrow P^{1\ldots u}$;
2: $H \leftarrow H|_{P_{\text{red}}}$, /* *restrict to* P_{red} */;
3: Compute $D_{\text{dps}}(q)$ for all $q \in P_{\text{red}}$;
4: Sort $P_{\text{del}} \leftarrow \{q \in P_{\text{red}} \mid D_{\text{dps}}(q) < 1\}$ in increasing $D_{\text{dps}}(q)$;
5: **if** $P_{\text{red}} \setminus \{p\}$ is feasible for each $q \in P_{\text{del}}$ **then**
6: $\quad P_{\text{red}} \leftarrow P_{\text{red}} \setminus \{q\}$;
7: **end if**
8: Return P_{red}.

$F_{D_{\text{dps}}}$.

III-6-3 Computational Complexity

In SFPS, the first step is to calculate cov_{dps} and sep_{dps}. As discussed in previous section, the computational complexity for those calculations is $O(m^2n)$. The computational complexity for calculating $F_{D\text{dps}}(P^{1\ldots u} \cup \{q\})$ is $O(m^2)$, given cov_{dps} and sep_{dps}. But this calculation takes $n - |P^{1\ldots u}|$ steps to find out q^+, when $q^+ \leftarrow \arg\max_{q \in P \setminus P^{1\ldots u}} F_{D_{\text{dps}}}(P^{1\ldots u} \cup \{q\})$. In conclusion, the computational complexity for SFPS is $O(m^2n^2)$ in the worst case.

CHAPTER IV

EVOLUTIONARY HEURISTICS FOR NON-UNIQUE PROBE SELECTION

Scientific discussion of evolution dates back than 200 years. Darwin suggested that slight variation among individuals significantly affects the gradual evolution of the population. This differential reproductive process of varying individuals is called natural selection. Evolutionary methods, which are inspired by the analogy of evolution and population genetics, are stochastic and optimization techniques. They have been demonstrated to be effective and robust in searching huge spaces in a wide range of applications. Evolutionary methods generally involve techniques implementing mechanisms such as reproduction, mutation, recombination (crossover), selection and survival of the fittest. Evolutionary methods usually are comprised of genetic algorithms (GAs), genetic programming (GP), evolutionary programming (EP) and evolution strategy (ES).

Genetic algorithms (GAs) are population based search algorithms. GAs became a widely recognized optimization method as a result of the work of *John Holland* in the early 1970s, and particularly his book in 1975. The individuals of population in a GA are usually represented as fixed length binary strings but there are GAs that use strings from higher cardinality alphabets and with variable length. Recombination (crossover) is the primary operator and mutation is considered as a secondary search operator.

Genetic programming (GP) is a form of evolutionary methods in which the indi-

viduals in the evolving population are computer programs rather than bit strings.

Evolutionary programming (EP) was originally conceived by *Lawrence J. Fogel* in 1966. Evolutionary programming is a stochastic optimization strategy similar to GAs. EP uses problem oriented representation. Mutation is the primary operator and depends on the representation used. It is usually adaptive, and crossover is rarely used.

Evolutionary strategy was invented by *Ingo Rechenberg* in 1960s and 70s. Initially ES used selection and mutation on one individual only. Recombination and larger populations were introduced later.

Non-unique probe selection problem is actually the constrained optimization problem. Some genetic algorithm approaches [1][2][22] have been proposed for the *set cover problem*, which is a similar constrained optimization problem. Penalty function methods have been the most popular approach to solve constrained optimization problems using genetic algorithms or evolutionary strategy, however the performance is not always satisfactory [7]. Another alternative is to design heuristic operators to transform infeasible solutions into feasible solutions [2]. The genetic algorithm and evolutionary strategy [42][43][44] presented in this thesis apply the heuristic feasibility operator based on our greedy heuristic research to solve the non-unique probe selection problem, and experiment results are comparable to those of the current state-of-the-art approaches.

IV-1 Genetic Algorithm with DRC Heuristic

This section discusses the proposed genetic algorithm for the non-unique probe selection problem, including the representation, fitness function, selection operator, crossover operator, mutation operator, heuristic feasibility operator, and population initialization and replacement strategy. Figure 7 describes the flow chart of the genetic algorithm proposed.

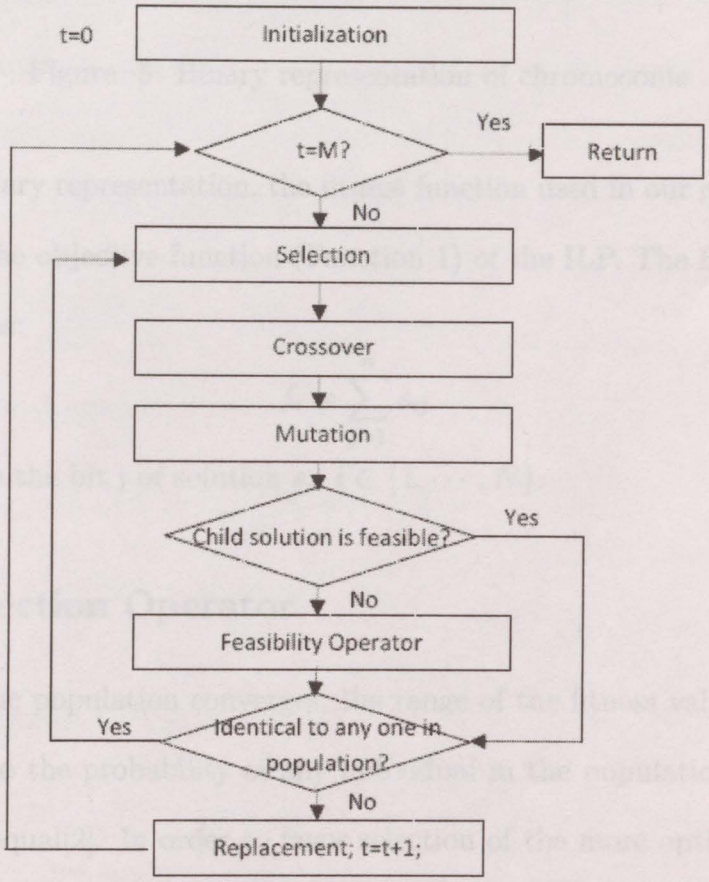


Figure 4: Flow chart of GA_DRC

IV-1-1 Representation and Fitness Function

The binary representation is an obvious choice for the non-unique probe selection problem here. We choose a n-bit binary string, shown in Figure 5, as the chromosome structure where n is the number of total probes. A value of 1 for the ith bit implies that probe p_i is in the solution.

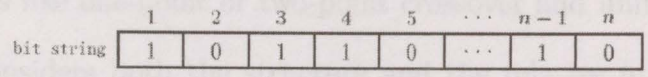


Figure 5: Binary representation of chromosome

With the binary representation, the fitness function used in our genetic algorithm coincides with the objective function (Function 1) of the ILP. The fitness function f is then defined as:

$$f_i = \sum_{j=1}^{n} s_{ij} \tag{40}$$

where $s_{ij} = x_j$ is the bit j of solution s_i, $i \in \{1, \cdots, N\}$.

IV-1-2 Selection Operator

Because when the population converges, the range of the fitness values in the population reduces, so the probability of any individual in the population to be selected become almost equal[2]. In order to favor selection of the more optimal individuals, we use fitness scaling and tournament selection. Fitness scaling maps an individual's raw fitness value onto a new value by subtracting a suitable value from the raw fitness as

$$f_i^s = f_i - min(f_i, i = 1, \cdots, N) \tag{41}$$

where f_i and f_i^s denote the raw fitness and the scaled fitness of individual i respectively, and N is the population size[2].

IV-1-3 Crossover Operator

Fusion operator is a generalized fitness-based crossover operator. Different with other crossover operators like one-point or two-point crossover and uniform crossover, the fusion operator considers both the structure and the relative fitness of the parent solutions, and produces just a single child instead of two children[2]. Let $f_{P_1}^s$ and $f_{P_2}^s$ be the scaled fitness value of the parent solutions P_1 and P_2 respectively, and let C denotes the child solution, then for all $i = 1, \cdots, n$:

1. if $P_1[i] = P_2[i]$, then $C[i] = P_1[i] = P_2[i]$;

2. if $P_1[i] \neq P_2[i]$, then

   - $C[i] = P_1[i]$ with probability $p = \frac{f_{P_2}^s}{f_{P_1}^s + f_{P_2}^s}$
   - $C[i] = P_2[i]$ with probability $1 - p$.

IV-1-4 Mutation Operator

Mutation works by inverting each bit in the solution with small probability and provides a small amount of random search[2]. The traditional genetic algorithms usually imply fixed mutation rate, but it is also suggested that $1/n$ as an optimal fixed mutation rate, where n is the length of the chromosome. A variable mutation schedule was considered because it is found that a higher mutation rate is preferred

when the GA has converged, and it is beneficial to utilize a variable mutation rate rather than a fixed one [2]. The number of bits mutated Num_{mut} can be defined as:

$$Num_{mut} = \lceil \frac{m_f}{1 + exp(-4m_g(t - m_c)/m_f)} \rceil \tag{42}$$

where t is the number of child solutions that have been generated,m_f specifies the final stable mutation rate, m_c specifies the number of child solutions generated at which a mutation rate of $m_f/2$ is reached and m_g specifies the gradient at $t = m_c$.

IV-1-5 Heuristic Feasibility Operator

The solutions generated by the crossover and mutation operators usually can not satisfy the problem constraints. So we propose a heuristic operator tailored specifically for the *non-unique probe selection problem* to maintain the feasibility of the solutions. The heuristic operator consists of two phases: "Construction Phase" and "Reduction Phase". In the construction phase, we initially start with a candidate set P_{sol} that is the unfeasible solution generated by the crossover and mutation operators. We then add probes into P_{sol} from $P - P_{sol}$ to generate the feasible solution. There maybe some redundant probes in P_{sol}, but they will be deleted during the reduction phase to generate a near minimal solution.

IV-1-6 Population Initialization and Replacement Strategy

Generally, the big population size is preferred such that the solution domain associated with the population is adequately covered. But sometime big population size is clearly too large for the GA to work efficiently, so we use specific initialization strategy

ALGORITHM 5 Construction Phase in Feasibility Operator of GA_DRC

Input: infeasible solution P_{sol}
Output: feasible solution P_{sol}

1: **for** each target t not covered by at least c_{min} probes **do**
2: **repeat**
3: add one probe p from $P - P_{sol}$ into P_{sol}, such that p hybridizes to t and has the highest $V(p)$
4: **until** the coverage constraint is satisfied for t.
5: **end for**
6: **for** each pair of targets not separated by at least h_{min} probes **do**
7: **repeat**
8: add one probe p from $P - P_{sol}$ into P_{sol}, such that p distinguish this pair of targets and has the highest possible value $V(p)$
9: **until** the separation constraint is satisfied for this target pair
10: **end for**

ALGORITHM 6 Reduction Phase in Feasibility Operator of GA_DRC

Input: P_{sol} with redundant probes
Output: P_{sol} without redundance

1: Update the incidence matrix H as $h_{ij} = 0$ for each $p_j \in P - P_{sol}$, $1 \leq i \leq m$, $1 \leq j \leq n$
2: Re-compute new C, S and V models from H
3: Set $P_{del} = \{\text{set of probes } p \in P_{sol} \mid v(p) < 1\}$ and sort P_{del} in increasing order
4: **repeat**
5: select p from P_{del} following the order
6: **if** $P_{sol} - \{p\}$ is feasible **then**
7: delete p from P_{sol}
8: **end if**
9: **until** every probe in P_{del} has been tried
10: Return final P_{sol}.

as following in order to work efficiently on relative smaller population size. The initialization of each solution s_i, $i \in \{1, \cdots, N\}$ in the population follow 2 steps:

- For each j from 1 to n, generate a random number $r \in [0, 1)$, then

$$s_{ij} = \begin{cases} 1 & \text{if } r \leq D_{drc}(p_j) \\ 0 & \text{else} \end{cases}$$

- If the solution is infeasible, then call heuristic feasibility operator.

After the initialization, all solutions in the population are feasible and ready for other genetic operators.

Once a new feasible child solution is generated, we apply the incremental replacement or steady-state replacement strategy that the child will replace a randomly chosen member which has an above average fitness value in the population. Here the above average fitness means less fit.

IV-1-7 Algorithms

Generally, the presented genetic algorithm can be summarized to the following steps (Algorithm 7).

IV-2 Evolution Strategy with DDRC and DDPS

In this section, we describe an Evolution Strategy(ES) that optimizes the solution obtained by our deterministic greedy methods. In computer science, ES is an

ALGORITHM 7 Genetic Algorithm with DRC Heuristic (GA_DRC)

Input: $T = \{t_1, \ldots, t_m\}$, $P = \{p_1, \ldots, p_n\}$, and $H = [h_{ij}]$
Output: Near-minimal solution P_{sol}

1: Generate an initial population of N solutions. Set $t := 0$.
2: **repeat**
3: Select two solutions P_1 and P_2 from the population using fitness scaling and binary tournament selection.
4: Produce a new solution C using the fusion crossover operator.
5: Mutate Num_{mut} randomly selected bits in C.
6: Make C feasible and remove redundant probes in C by using the heuristic feasibility operator.
7: **if** C is identical to any one of the solutions in the population **then**
8: go to step 3;
9: **else**
10: set $t := t + 1$ and go to step 12;
11: **end if**
12: Replace a randomly selected solution with an above-average fitness in the population by C.
13: **until** $t = M$ non-duplicate solutions have been generated

ALGORITHM 8 Evolution Strategy with DDRC Heuristic (DDRC_ES)

Input: $T = \{t_1, \ldots, t_m\}$, $P = \{p_1, \ldots, p_n\}$, and $H = [h_{ij}]$
Output: Near-minimal solution $P_{\min}$

1: $P_{min} \leftarrow DDRC(P, T, H)$
2: **repeat**
3: **repeat**
4: $P_{mut} \leftarrow Mutation(P_{min}, P)$
5: $P_{con} \leftarrow Construction(P_{mut}, P, T, H)$
6: $P_{red} \leftarrow Reduction(P_{con}, P, T, H)$
7: **if** $|P_{red}| < |P_{min}|$ **then**
8: $P_{min} \leftarrow P_{red}$
9: **end if**
10: **until** n_{gen} generations are performed
11: **until** n_{ite} iterations are performed
12: Return final P_{min}

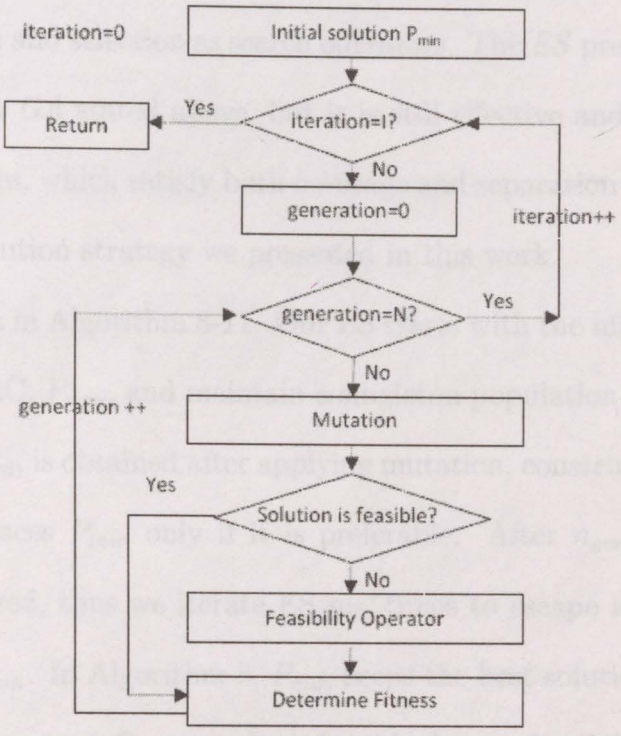


Figure 6: Flow chart of ES

ALGORITHM 9 Mutation in DDRC_ES

Input: P_{min}, $P = \{p_1, \ldots, p_n\}$

Output: P_{mut}

1: $P_{mut} \leftarrow P_{min}$
2: Generate a random number $r \in [1, |p|]$
3: **repeat**
4: Randomly select a probe $p \in P$
5: **if** $p \in P_{mut}$ **then**
6: $P_{mut} \leftarrow P_{mut} \setminus \{p\}$ with probability $1 - D_{drc}(p)$
7: **else**
8: $P_{mut} \leftarrow P_{mut} \cup \{p\}$ with probability $D_{drc}(p)$
9: **end if**
10: **until** r probes are processed
11: Return final P_{mut}

optimization technique based on ideas of evolution. Usually evolution strategies primarily use mutation and selection as search operators. The *ES* presented in this thesis is more simple than *GA* stated above, but it is still effective and robust to search a smallest set of probes, which satisfy both coverage and separation constraints. Figure 6 describes the evolution strategy we presented in this work.

Our ES is shown in Algorithm 8-11. Our ES starts with the initial parent solution obtained from DDRC, P_{min}, and maintain a singleton population in each generation. A child solution, P_{red}, is obtained after applying mutation, construction and reduction on P_{min}. P_{red} replaces P_{min} only if it is preferable. After n_{gen} generations, P_{min} may not be optimized, thus we iterate ES n_{ite} times to escape local optima and to further optimize P_{min}. In Algorithm 8, P_{min} keeps the best solution so far. However, after mutation, the mutant P_{mut} may be infeasible; hence, feasibility operator will be applied in order to generate a feasible near-minimal solution.

Here we can definitely use the similar heuristic feasibility operator, which consists of *construction* and *reduction* phases, with different calculation of $D(p)$, compared with that in GA_DRC.

ALGORITHM 10 Construction in DDRC_ES

Input: P_{mut}, P, T, H

Output: Feasible solution P_{con}

1: $P_{con} \leftarrow P_{mut}$
2: **for** each target t_i not c_{min} covered by P_{con} **do**
3: $n_i \leftarrow$ # probes needed to complete c_{min}-coverage of t_i
4: **repeat**
5: $P_{con} \leftarrow P_{con} \cup \{q \in P \setminus P_{con}$ with highest degree that covers $t_i\}$
6: **for** all $t_a (1 \leq a \leq m)$ and $t_{ab} (1 \leq a < b \leq m)$ covered by q **do**
7: Update $D(p)$ for all $p \in \{P_{t_a} \setminus C_{t_a}\} \cup \{P_{t_{ab}} \setminus S_{t_{ab}}\}$
8: **end for**
9: $H \leftarrow H|P \setminus \{q\}$
10: $P \leftarrow P \setminus \{q\}$
11: **until** n_i probes are inserted
12: **end for**
13: **for** each target pair t_{ik} not s_{min} separated by P_{con} **do**
14: $n_{ik} \leftarrow$ # probes needed to complete s_{min} separation of t_{ik}
15: **repeat**
16: $P_{con} \leftarrow P_{con} \cup \{$ probe $q \in P \setminus P_{con}$ with highest degree that separate $t_{ik}\}$
17: **for** all $t_a (1 \leq a \leq m)$ and $t_{ab} (1 \leq a < b \leq m)$ covered by q **do**
18: Update $D(p)$ for all $p \in \{P_{t_a} \setminus C_{t_a}\} \cup \{P_{t_{ab}} \setminus S_{t_{ab}}\}$
19: **end for**
20: $H \leftarrow H|P \setminus \{q\}$
21: $P \leftarrow P \setminus \{q\}$
22: **until** n_{ik} probes are inserted
23: **end for**
24: Return P_{con}

ALGORITHM 11 Reduction in DDRC_ES

Input: P_{con}, P, T, H

Output: Reduced solution P_{red}

1: $P_{red} \leftarrow P_{con}$
2: $H \leftarrow G|P_{red}$ {we restore initial H and restrict to P_{red}}
3: Compute $D(p) = D_{drc}(p)$ for all $p \in P_{red}$
4: Sort $P_{del} \leftarrow \{p \in P_{red} | D(p) < 1\}$ in increasing order
5: **if** $P_{red} \setminus \{p\}$ is feasible for each $p \in P_{del}$ **then**
6: $P_{red} \leftarrow P_{red} \setminus \{p\}$
7: **end if**
8: Return P_{red}

CHAPTER V

COMPUTATIONAL EXPERIMENTS

V-1 Data Description

Two groups of data have been used in the experiments. In this work, we assume that the initial candidate probe set is feasible. If not, we insert a sufficient number of unique virtual probes into P. For each target t_i or target-pair t_{ik} that a constraint is not satisfied, $(c_{min} - |P_{t_i}|)$ or $(s_{min} - |P_{t_{ik}}|)$ virtual unique probes are added.

V-1-1 Artificial Data Set

In order to evaluate the benefits of our methods more systematically, in our experiments, we also use the artificial data sets, which was first described in [18], and have already been used in [30][19][23][25].

To generate artificial data that closely models homologous sequence families, Klau *et al.* [18] use the REFORM (Random Evolutionary FORests Model) software that allows to define arbitrary sets of evolutionary trees. Two different forest models were used, and for each model, five independent test sets were generated [18]. A family of 256 sequences of average length 1000nt are produced for the first model. In the second model, all global parameters are same as in the first model, and the sequences consist of a single segment of average length 1000 nt, but the topology differs considerably from the the first model.

Promide software were used to generate probe candidates for each of the 10 fam-

ilies. Probe candidates are selected to be between 19 and 21 nt long and have a stability (Gibbs energy) of -20 to -19.5 kcal/mol at $40°C$ and 0.075 M $[Na^+]$ according to the Nearest Neighbor model [18]. More details of those artificial data sets can be found in [18]. Table 7 describes the number of targets and probes for artificial data sets used in experiments. $|A|$ denotes the virtual unique probes added to make each candidate probe set is feasible.

Table 7: Artificial data set

Set	$\|T\|$	$\|P\|$	$\|A\|$
a1	256	2786	6
a2	256	2821	2
a3	256	2871	16
a4	256	2954	2
a5	256	2968	4
b1	400	6292	0
b2	400	6283	1
b3	400	6311	5
b4	400	6223	0
b5	400	6285	3

V-1-2 Real Data Set

The real data group consists of a set of 28S rDNA sequences from different organisms present in the Meiobenthos, HIV-1 data set and HIV-2 data set.

To reduce the level of redundancy of original 1230 28S rDNA sequences of Meiobenthos, Schliep et al.[32] used the blastclust software from NCBI to cluster sequences in the data set, and selected arbitrary representatives of all sequences. As a result, the test set consists of 679 sequences. The HIV-1 and HIV-2 sequences were chosen

in particular because of their biological significance and because the sequences were very closely related and similar within each set. This made them good candidates for the non-unique probe selection problem. Two hundred sequences of each type were downloaded from NCBI (the National Center for Biotechnology Information). Candidate probes for the sequences were generated using Primer3 with default parameters, which included: length between 18 and 27 nucleotides, melting temperature between 57 and 63 , and GC content between 20 and 80%. 40 probes for each sequence were generated for each data set, and duplicate probes were deleted before the target-probe incident matrix was constructed. Table 8 details the number of targets and probes for M, HIV-1 and HIV-2 data set used in experiments.

Table 8: Real data set

Set	$\|T\|$	$\|P\|$	$\|A\|$
M	679	15139	75
HIV-1	200	4806	20
HIV-2	200	4686	35

V-2 Experiment Parameters and Results

We performed experiments to show the minimization ability of heuristics presented in this thesis. All programs were written in C and all tests ran on two Intel Xeon^{TM} CPUs 3.60GHz with 3GB of RAM under Ubuntu 6.06 i386.

All experiments were done with parameters $c_{min} = 10$ and $s_{min} = 5$.

V-2-1 Experiment Results of Deterministic Greedy Heuristics

Table 9 shows, for all data sets, the minimum sizes $|P_{min}|$ attained by the greedy methods, DRC, DPS, DPSn, DDRC, DDPS, DDPSn and SFPS. Table 10 shows the

Table 9: Computational results of deterministic greedy heuristics

Set	$\|P\|+\|A\|$	DRC	DPS	DPSn	DDRC	DDPS	DDPSn	SFPS
a1	2792	549	547	547	523	519	511	530
a2	2823	552	537	526	510	502	501	516
a3	2887	590	577	573	543	544	542	557
a4	2956	579	578	580	552	548	547	557
a5	2972	583	571	564	551	543	537	558
b1	6292	974	921	924	884	880	875	883
b2	6284	1013	942	970	892	887	880	890
b3	6316	953	915	923	879	881	868	896
b4	6223	1019	956	973	919	905	905	920
b5	6288	1019	969	987	929	918	921	933
M	15214	2084	2068	2061	1996	2016	1986	2036
HIV-1	4826	487	472	476	459	461	460	468
HIV-2	4721	506	501	501	487	488	487	492

running time of each greedy heuristic for all data sets.

V-2-2 Experiment Parameters and Results of GA_DRC

In the approach GA_DRC (Section IV-1), the population size N was set to 100, m_f was set to 10, m_c was set to 200 and m_g was set to 2.0 for all the datasets (these values were obtained by trial-and-error). We ran GA_DRC ten times on each data set with different random seed. Each run terminated when $M = 10,000$ non-duplicate solutions had been generated. Figure 7 shows the comparison on dataset $b1$ among

Table 10: Running time of deterministic greedy heuristics

Set	DRC (s)	DPS (s)	DPSn (s)	DDRC (s)	DDPS (s)	DDPSn (s)	SFPS (s)
a1	2	4	4	7	6	539	347
a2	3	4	4	7	7	556	341
a3	3	5	4	7	7	684	372
a4	3	4	4	7	7	782	401
a5	3	4	4	7	7	688	382
b1	16	22	18	37	36	4039	4120
b2	15	21	18	36	37	4028	4231
b3	16	21	18	37	36	5074	4006
b4	14	20	18	36	35	3997	4040
b5	14	21	19	37	36	4176	4339
M	78	140	130	277	315	46318	37546
HIV-1	2	4	4	7	8	635	338
HIV-2	3	4	3	6	6	620	353

Table 11: Computational results of genetic algorithm with DRC

Set	$\|P\| + \|A\|$	Min	Ave	Max	Time (h)
a1	2792	502	503.9 ± 1.3	506	2.07 ± 0.05
a2	2823	490	491.4 ± 0.7	492	2.08 ± 0.03
a3	2887	534	534.8 ± 1.0	537	2.21 ± 0.06
a4	2956	537	538.2 ± 0.6	539	2.01 ± 0.04
a5	2972	528	528.2 ± 0.4	529	2.15 ± 0.02
b1	6292	839	842.2 ± 2.0	845	7.41 ± 0.13
b2	6284	852	854.8 ± 2.0	859	7.43 ± 0.12
b3	6316	835	838.7 ± 2.5	842	7.44 ± 0.18
b4	6223	879	882.5 ± 3.0	889	7.39 ± 0.1
b5	6288	890	892.8 ± 2.4	897	7.39 ± 0.08
M	15214	1962	1964.3 ± 2.5	1971	62.29 ± 2.75
HIV-1	4826	450	450.7 ± 0.5	451	1.38 ± 0.02
HIV-2	4721	476	477.7 ± 0.8	479	1.32 ± 0.01

the genetic algorithm (GA_DRC) presented in Section IV-1 (denoted as GA1) and GA2 that use totally random population initialization instead of the initialization strategy described in Section IV-1-6. In this figure, we found that before 4170 children generated, GA2 performed better than GA1; while with more children generated, GA1 kept the optimal solutions.

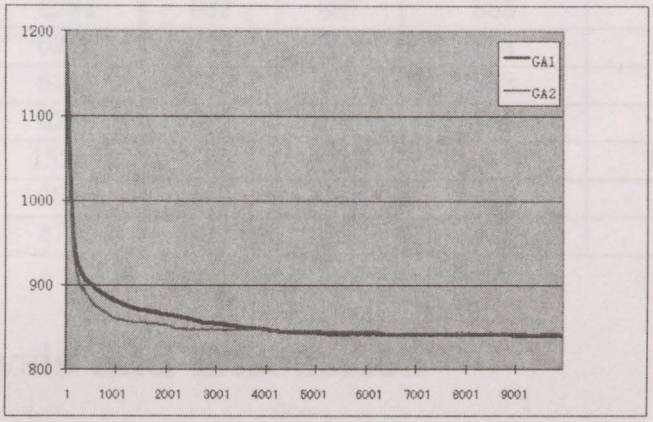


Figure 7: Comparison of GAs

V-2-3 Experiment Parameters and Results of ES

For evolution strategy, the values for parameters (n_{ite}, n_{gen}) were (100, 100) for DDRC_ES and (1, 100) for DDPS_ES respectively. DDRC_ES terminated in two weeks given all the thirteen data sets altogether. The parameters values for DDPS_ES were determined such that it terminates in two weeks. Table 12 shows, for all data sets, the minimum sizes $|P_{min}|$ attained by the greedy methods, DDRC and DDPS, and the evolution strategy, DDRC_ES and DDPS_ES. It is easy to see that DDRC_ES and DDPS_ES substantially outperformed DDRC and DDPS in all instances.

Table 12: Computational results of evolution strategy

Set	$\|P\| + \|A\|$	DDRC	DDPS	DDRC_ES	DDPS_ES
a1	2792	523	519	506	505
a2	2823	510	502	494	490
a3	2887	543	544	535	536
a4	2956	552	548	539	540
a5	2972	551	543	531	529
b1	6292	884	880	857	866
b2	6284	892	887	865	873
b3	6316	879	881	854	864
b4	6223	919	905	888	900
b5	6288	929	918	905	911
M	15214	1996	2016	1972	1996
HIV-1	4826	459	461	452	457
HIV-2	4721	487	488	478	479

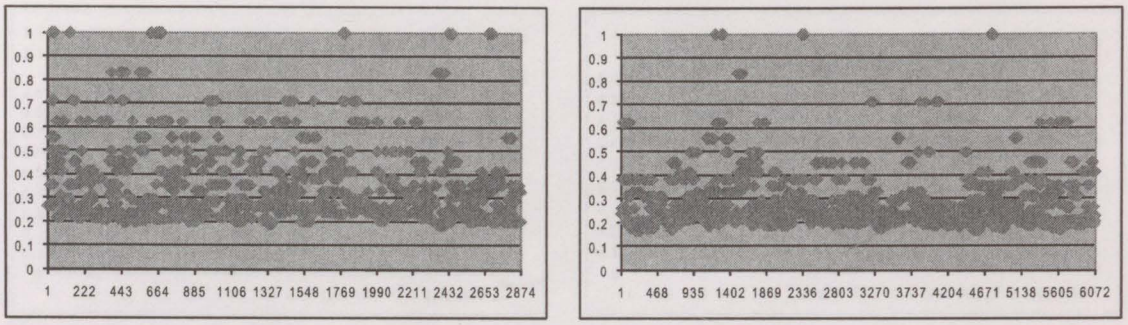
V-3 Analysis and Discussion

Table 13 shows, for all data sets, the minimum sizes $|P_{min}|$ and the percentages in relation to the number of probe candidates, attained by all approaches proposed in this thesis, the greedy heuristics of [32] (GrdS) and [23] (GrdM), the Integer Linear Programming (ILP) [18][19], and the optimal cutting-plane algorithm (OCP) [25]. Given two heuristics X and Y, we say that $X < Y$ in terms of their overall performances on the data sets, if X produces larger solutions than Y in the majority of the data sets. From Table 13, we can see the following order: $GrdS < GrdM < DRC < DPSn < DPS < SFPS < DDRC < ILP < DDPS < DDPSn < DDPS\_ES < DDRC\_ES < OCP < GA\_DRC$.

The GrdM [23] heuristic sorts the probes in decreasing order of the number of targets they hybridize, then selects probes in this order to satisfy the constraints,

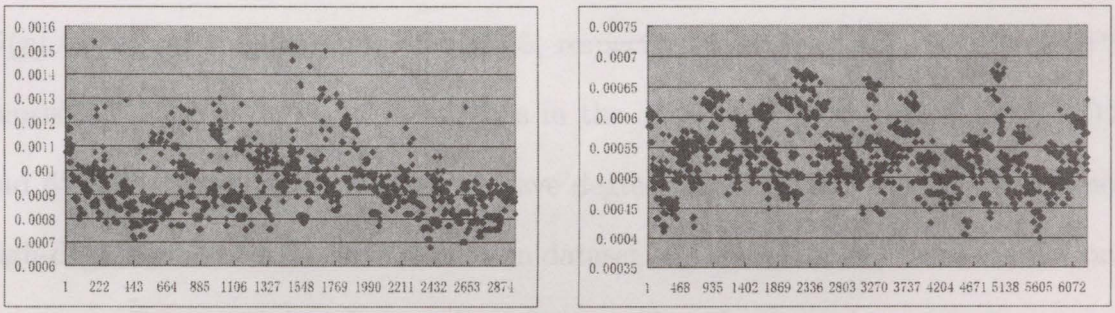
Table 13: Experiment results overview

Set	DRC	DPS	DPSn	DDRC	DDPS	DDPSn	SFPS	DDRC_ES	DDPS_ES	GA_DRC	GrdS	GrdM	ILP	OCP
a1	549 (19.66%)	547 (19.59%)	547 (19.59%)	523 (18.73%)	519 (18.59%)	511 (18.3%)	530 (18.98%)	506 (18.12%)	505 (18.09%)	502 (17.98%)	1163 (41.65%)	568 (20.34%)	503 (18.02%)	509 (18.23%)
a2	552 (19.55%)	537 (19.02%)	526 (18.63%)	510 (18.07%)	502 (17.78%)	501 (17.75%)	516 (18.28%)	494 (17.5%)	490 (17.36%)	490 (17.36%)	1137 (40.28%)	560 (19.84%)	519 (18.38%)	494 (17.5%)
a3	590 (20.44%)	577 (19.99%)	573 (19.85%)	543 (18.81%)	544 (18.84%)	542 (18.77%)	557 (19.29%)	535 (18.53%)	536 (18.57%)	534 (18.5%)	1175 (40.7%)	613 (21.23%)	516 (17.87%)	543 (18.81%)
a4	579 (19.59%)	578 (19.55%)	580 (19.62%)	552 (18.67%)	548 (18.54%)	547 (18.5%)	557 (18.84%)	539 (18.23%)	536 (18.27%)	534 (18.17%)	1169 (39.55%)	597 (20.2%)	540 (18.27%)	539 (18.23%)
a5	583 (19.62%)	571 (19.21%)	564 (18.98%)	551 (18.54%)	543 (18.27%)	537 (18.07%)	558 (18.78%)	531 (17.87%)	529 (17.8%)	528 (17.77%)	1175 (39.54%)	605 (20.36%)	504 (16.96%)	529 (17.8%)
b1	974 (15.48%)	921 (14.64%)	924 (14.69%)	884 (14.05%)	880 (13.99%)	875 (13.91%)	883 (14.03%)	857 (13.62%)	866 (13.76%)	839 (13.33%)	1908 (30.32%)	961 (15.27%)	879 (13.97%)	830 (13.19%)
b2	1013 (16.12%)	942 (14.99%)	970 (15.44%)	892 (14.19%)	887 (14.12%)	880 (14%)	890 (14.16%)	865 (13.77%)	873 (13.89%)	852 (13.56%)	1885 (30%)	976 (15.53%)	938 (14.93%)	842 (13.4%)
b3	953 (15.09%)	915 (14.49%)	923 (14.61%)	879 (13.92%)	881 (13.95%)	868 (13.74%)	896 (14.19%)	854 (13.52%)	864 (13.68%)	835 (13.22%)	1895 (30%)	951 (15.06%)	891 (14.11%)	827 (13.09%)
b4	1019 (16.37%)	956 (15.36%)	973 (15.64%)	919 (14.77%)	905 (14.54%)	905 (14.54%)	920 (14.78%)	888 (14.27%)	900 (14.46%)	879 (14.13%)	1888 (30.34%)	1001 (16.09%)	915 (14.7%)	873 (14.03%)
b5	1019 (16.21%)	969 (15.41%)	987 (15.7%)	929 (14.77%)	918 (14.6%)	921 (14.65%)	933 (14.84%)	905 (14.39%)	911 (14.49%)	890 (14.15%)	1876 (29.83%)	1022 (16.25%)	946 (15.04%)	874 (13.9%)
M	2084 (13.7%)	2068 (13.59%)	2061 (13.55%)	1996 (13.12%)	2016 (13.25%)	1986 (13.05%)	2036 (13.38%)	1972 (12.96%)	1996 (13.12%)	1962 (12.9%)	3851 (25.31%)	2336 (15.35%)	3158 (20.76%)	1962 (12.9%)
HIV1	487 (10.09%)	472 (9.78%)	476 (9.86%)	459 (9.51%)	461 (9.55%)	460 (9.53%)	468 (9.7%)	452 (9.37%)	457 (9.47%)	450 (9.32%)	-	531 (11%)	-	451 (9.35%)
HIV2	506 (10.72%)	501 (10.61%)	501 (10.61%)	487 (10.32%)	488 (10.34%)	487 (10.32%)	492 (10.42%)	478 (10.13%)	479 (10.15%)	476 (10.08%)	-	578 (12.24%)	-	479 (10.15%)



(a) Distribution in a5 (b) Distribution in b5

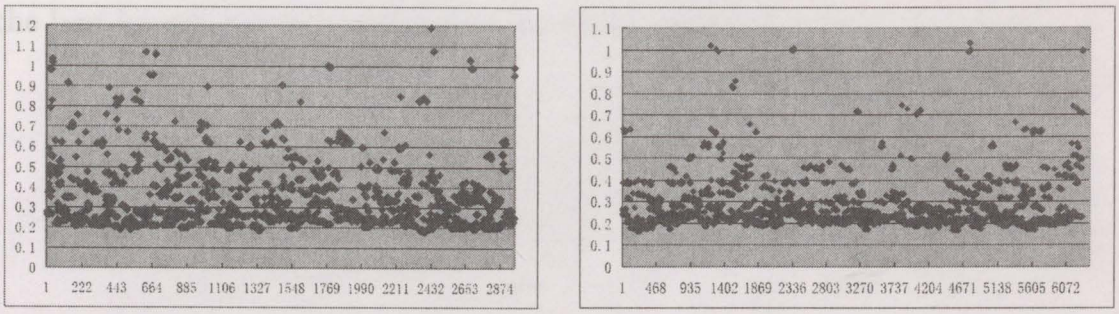
Figure 8: DRC's $D(p)$ distribution in (a) the a5 data set and (b) the b5 data set



(a) Distribution in a5 (b) Distribution in b5

Figure 9: DPS's $D(p)$ distribution in (a) the a5 data set and (b) the b5 data set

and finally tries to remove redundant probes randomly. This probe sorting process is similar to selecting dominant probes, though it is not encoded in a selection function. GrdM uses no other information or any selection function, thus it cannot identify the quality of probes that hybridize to the same number of targets. Also, selecting only dominant probes does not guarantee that dominated targets are covered earlier in the selection process, and therefore GrdM yields larger solutions than ours. Our heuristics encode useful information about each probe in a selection function, and are able to identify good probes. In Figure 8- 10, we show the distribution of the initial



(a) Distribution in a5

(b) Distribution in b5

Figure 10: DPSn's $D(p)$ distribution in (a) the a5 data set and (b) the b5 data set

$D(p)$ values for the data sets $a5$ and $b5$, respectively for the DRC, DPS and DPSn heuristics. In DRC, 37% of the probes in the dataset $a5$ have degree $D(p) < 0.25$ and 65% of probes in the dataset $b5$ have degree $D(p) < 0.25$. Also there are more high-degree probes in dataset $a5$ than in dataset $b5$. In dataset $b5$, there are not only too many probes with low degrees, but also many low-degree probes with almost same values $D(p)$. DRC does not encode enough information to select between them, so for such datasets, GrdM performs better than DRC by selecting the dominant probes among these similar probes.

Compared with the OCP heuristic, DDRC, DDPS and dDPSn heuristics produced results that are within at most 6.5%, 6.5%, and 5.4% of the results of OCP. This is quite good given that these are only simple greedy methods plus being faster than OCP. In particular, the mean improvements of DDPS and DDPSn relative to OCP are +3.2 and +2.6 respectively, which are very low and hence very good.

Our evolution strategy approaches also produced near-optimal results that are very close to those of OCP, and meanwhile, genetic algorithm with DRC obtained

the best known optimal solutions for 6 over 13 instances.

CHAPTER VI

CONCLUSION

VI-1 Summary of Contributions

In this thesis, the sequential forward search algorithm, genetic algorithm and evolution strategy are applied for the first time to solve the minimization problem arisen from the non-unique probe selection, respectively. Currently, we just consider the case for single target separations only, not aggregated target set separations.

Compared with the state-of-the-art heuristics, DDRC, DDPS and DDPSn heuristics produced results close to those of OCP, using the same datasets. This is quite good given that these are only simple greedy methods beside being faster than OCP. This suggests that more powerful heuristics that make use of selection functions would give better overall performance than OCP.

Meanwhile, the results showed that the first evolution strategy approaches (DDRC_ES and DDPS_ES) for the non-unique probe selection problem, presented in this work, are able to obtain results that are very close to those of OCP, and the genetic algorithm with DRC obtained a better overall performance than OCP.

The selection functions presented in this thesis, can be modified to be used in well-known problems in bioinformatics and computational biology that are expressed as minimal set covering problems, like protein-protein interaction prediction, oligonucleotide primer design and siRNA selection for RNA interference experiments.

VI-2 Future Work

SFPS outperformed some published greedy algorithms and gave results close to the optimal search method of ILP, but SFPS also suffers from the *nesting effect* of SFS; that is, a probe that was selected cannot be discarded later to correct a wrong decision, and hence the solution tends to be sub-optimal. The main cause of the nesting effect is the use of a monotonic criterion such as our $F_{D_{dps}}$ criterion. Other sequential methods, such as the floating search methods [24], will be good choice to reduce the nesting effect and cope with non-monotonic criterion functions.

Experiments showed that evolutionary methods proposed are able to obtain near minimal solutions comparable to the best known methods for this problem. But the running time of those evolutionary methods shows them not very practical. However, since the probe set for microarray is only created once, the time spent to compute the minimal probe set is far less crucial than the size and quality of the probe set. Further improvements can be applied to speed them up by using parallel computing techniques.

As we currently focus on the computation of the of the minimum set of candidate probes with the minimum coverage and separation constraints, given a target set T, probe set P, and the target-probe incidence matrix H, based on the ILP formulation (Equation 1) without group separation constraints, and provide practical algorithms rather than theoretical analysis, the group separation constraints can be considered as extension of those algorithms in future.

REFERENCES

[1] Aickelin, U. 2002. An indirect genetic algorithm for set covering problems. *J. Oper. Res. Soc. 53*, 1118-1126.

[2] Beasley, J. E. and Chu, P. C. 1996. A genetic algorithm for the set covering problem, *European J. Oper. Res. 94*, 392-404.

[3] Borneman, J., Chroback, M., Vedova, G.D. Figueroa, A., and Jiang, T. 2001. Probe selection algorithms with applications in the analysis of microbial communities. *Bioinformatics 17*, Suppl 1:s39-s48.

[4] Cazalis, Z., Milledge, T., and Narasimhan, G. 2004. Probe selection problem: structure and algorithms. In *Proceedings of the 8th Multi-Conference on Systemics, Cybernetics and Informatics (SCI2004)*, 124-129.

[5] Couzinet, S., Jay, C., Barras, C., Vachon, R., Vernet, G., Ninet, B., Jan, I., Minazio, M.A., Francois, P., Lew, D., Troesch, A., and Schrenzel, J. 2004. High-density DNA probe arrays for identification of staphylococci to the species level. *Journal of Microbiological Methods 61*, 2, 201-208.

[6] Cutichia, A., Arnold, J., and Timberlake, W. 1993. PCAP: probe choice and analysis package - a set of programs to aid in choosing synthetic oligomers for contig mapping. *CABIOS 9*, 201-203.

[7] Deb, K. 2000. An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering 186*, 2-4, 311-338.

[8] Deng, P., Wang, F., and Du, D.Z. 2007. Non-unique probe selection with group testing. In *Proceedings of the First International Symposium on Optimization and Systems Biology (OSB'07)*, Beijing, China. 1-4.

[9] Deng, P., Thai, M.T., Ma, Q., and Wu, W. 2008. Efficient non-unique probes selection algorithms for DNA microarray. *BMC Genomics 9*, Suppl 1:S22

[10] Diamandis, E.P. 2000. Sequencing with microarray technology- a powerful new tool for molecular diagnostics. *Clinical Chemistry 46*, 10, 1523-1525.

[11] Fu, L., Borneman, J., Ye, J., and Chrobak, M. 2005. Improved probe selection for DNA arrays using nonparametric kernel density estimation. In *Proceedings of the IEEE Engineering in Medicine and Biology 27th Annual Conference*, Shanghai, China.

[12] Gąsieniec, L., Li, C.Y., Sant, P., and Wong, P.W.H. 2006. Efficient Probe Selection in Microarray Design. In *Proceedings of IEEE Symposium on Computational Intelligence and Bioinformatics and Computational Biology (CIBCB'06)*, Toronto, Ontario, Canada.

[13] Gerhold, D., Rushmore, T., and Caskey, C.T. 1999. DNA chips: promising toys have become powerful toolds. *Trends. Biochem. Sci. 24*, 168-173.

[14] Goldberg, D. 1989, *Genetic Algorithm in Search, Optimization and Machine Learning*, Addison-Wesley, New York.

[15] Herwig, R., Schmitt, A.O., Steinfath, M., O'Brien, J., Seidel, H., Meier-Ewert, S., Lehrach, H., and Radelof, H. 2000. Information theoretical probe selection for hybridisation experiments. *Bioinformatics 16*, 10, 890-898.

[16] Huang, Y., Chang, C., Chan, C., Yeh, T., Chang, Y., Chen, C., and Kao, C. 2005. Integrated minimum-set primers and unique probe design algorithms for differential detection on symptom-related pathogens. *Bioinformatics 21*, 4330-4337.

[17] Kaderali, L. and Schliep, A. 2002. Selecting signature oligonucleotide to identify organisms using DNA arrays. *Bioinformatics 18*, 10, 1340-1349.

[18] Klau, G.W., Rahmann, S., Schliep, A., Vingron, M., and Reinert, K. 2004. Optimal robust non-unique probe selection using integer linear programming. *Bioinformatics 20*, i186-i193.

[19] Klau, G.W., Rahmann, S., Schliep, A., Vingron, M., and Reinert, K. 2007. Integer linear programming approaches for non-unique probe selection. *Discrete Applied Mathematics 155*, 840-856.

[20] Li, F. and Stormo, G. 2000. Selecting optimum DNA oligos for microarrays. In *Proceedings of IEEE International Symposium on Bio-Informatics and Biomedical Engineering (BIBE)*, Key Bridge Marriott, Arlington, USA.

[21] Lockhart, D.J., Dong, H., Byrne, M.C., Follettie, M.T., Gallo, M.V., Chee, M.S., Mittmann, M., Wang, C., Kobayashi, M., Horton, H., and Brown, E.L. 1996.

Expression monitoring by hybridization to high-density oligonucleotide arrarys. *Nature Biotechnology 14*, 13, 1675-1680.

[22] Lorena, L.A.N. and Lopes, L.S. 1997. Genetic algorithm applied to computationally difficult set covering problems. *Journal of the Operational Research Society 48*, 4, 440-445.

[23] Meneses, C.N., Pardalos, P.M., and Ragle, M.A. 2007. A new approach to the non-unique probe selection problem. *Annals of Biomedical Engineering 35*, 4, 651-658.

[24] Pudil, P., Ferri, F.J., Novovicova, J., and Kittler, J. 1994. Floating search methods for feature selection with nonmonotonic criterion functions. In *Proceedings of IAPR 12th International Conference on Pattern Recognition, Oct. 9-13*, Jerusalem, Israel, vol.2, 279-283.

[25] Ragle, M.A., Smith, J.C., and Pardalos, P.M. 2007, An optimal cutting-plane algorithm for solving the non-unique probe selection problem. *Annals of Biomedical Engineering 35*, 11, 2023-2030.

[26] Rahmann, S. 2002. Rapid large-scale oligonucleotide selection for microarrays. In *Proceedings of the First IEEE Computer Society Bioinformatics Conference(CSB)*, 54-63, Standford, CA, USA.

[27] Rahmann, S. 2003. Fast large-scale oligonucleotide selection using the longest common factor approach. *Journal of Bioinformatics and Computational Biology 1*, 2, 343-361.

[28] Rahmann, S. 2003. Fast and sensitive probe selection for DNA chips using jumps in matching statistics. In *Proceedings of of the 2nd IEEE Computer Society Bioinformatics Conference (CSB'03)*, Standford, CA, USA.

[29] Rahmann, S., M uller, T., and Vingron, M. 2004. Non-unique probe selection by matrix condition optimization. In *Currents in Computational Molecular Biology*, San Diego, USA.

[30] Rahmann, Sven 2004. Algorithms for Probe Selection and DNA Microarray Design. Dissertation. Max Planck Institue for Molecular Genetics, Berlin.

[31] Rash, S. and Gusfield, D. 2002. String barcoding: Uncovering optimal virus signatures. In *Proceedings of the Sixth Annual International Conference on Computational Biology*, Washington, DC, 254-261.

[32] Schliep, A., Torney, D.C., and Rahmann, S. 2003. Group testing with DNA chips: generating designs and decoding experiments. In *Proceedings of the 2nd IEEE Computer Society Bioinformatics Conference (CSB'03)*, Standford, CA, USA.

[33] Schliep, A. and Rahmann, S. 2006. Decoding non-unique oligonucleotide hybridization experiments of targets related by a phylogenetic tree. *Bioinformatics 22*, e424-e430.

[34] Shin, S.Y., Lee, I.H., and Zhang, B.T. 2006. Microarray probe design using ε-multi-objective evolutionary algorithms with thermodynamic criteria. *LNCS 3907*, 184-195.

[35] Snustad, D.P. and Simmons, M.J. 1999. *Principles of Genetics*, 2nd Edition, Wiley, New York.

[36] Sung, W.K. and Lee, W.H., 2003. Fast and accurate probe selection algorithm for large genomes, In *Proceedings of of the 2nd IEEE Computer Society Bioinformatics Conference (CSB'03)*, Standford, CA, USA.

[37] Thai, M., MacCallum, D., Deng, P., and Wu, W. 2007. Decoding algorithms in pooling designs with inhibitors and error-tolerance. *Int. J. Bioinformatics Research and Applications 3*, 2,145-152.

[38] Tobler, J.B., Molla, M.N., Nuwaysir, E.F. Green R.D., and Shavlik, J.W. 2002. Evaluating machine learning approaches for aiding probe selection for gene-expression arrays. *Bioinformatics 18*, s164-s171.

[39] Tulpan, D.C. 2006. Effective heuristic methods for DNA strand design. Dissertation. The University of British Columbia, Canada.

[40] Wang, F., Du, H., Jia,X., and Deng, P. 2007. Non-unique probe selection and group testing. *Theoretical Computer Science 381*, 1-3, 29-32.

[41] Wang, L. and Ngom, A. 2007. A model-based approach to the non-unique oligonucleotide probe selection problem, In *Proceedings of the Second International Conference on Bio-Inspired Models of Network, Informatiaon, and Computing Systems(Bionetics 2007)*, Dec.10-13, Budapest, Hungary, ISBN:978-963-9799-05-9.

[42] Wang, L., Ngom, A., and Gras, R. 2008. Non-unique oligonucleotide microarray probe selection method based on genetic algorithm. In *Proceedings of the 2008 IEEE Congress on Evolutionary Computation*, Jun. 1-6, Hong Kong, China, 1004-1011.

[43] Wang, L., Ngom, A., Gras, R., and Rueda, L. 2008. An evolutionary approach to the non-unique oligonuleotide probe selection problem, *Springer Transactions on Computational System Biology*, in press.

[44] Wang, L., Ngom, A., Gras, R., and Rueda, L. 2008. Evolution strategy with greedy probe selection heuristics for the non-unique oligonucleotide probe selection problem. In *Proceedings of the 2008 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, Sep.15-17, Sun Valley, Idaho, USA.

[45] Wang, L., Ngom, A., and Rueda, L. 2008. Sequential forward selection approach to the non-unique oligonucleotide probe selection problem. In *Proceedings of the third IAPR International Conference on Pattern Recognition in Bioinformatics*, Melbourne, Australia.

[46] Wang, X. and Seed, B. 2003. Selection of oligonucleotide probes for protein coding sequences. *Bioinformatics 19*, 7, 796-802.

[47] Wikipedia: http://en.wikipedia.org/wiki/Functional_genomics

VITA AUCTORIS

NAME: Lili Wang

PLACE OF BIRTH: Shandong, China

YEAR OF BIRTH: 1980

EDUCATION: University of Windsor

Windsor, Ontario, Canada

2006-2008 M.Sc.

Academy of Armoured Forces Engineering

Beijing, China

1998-2002 B.Sc.